**Write-up for Project 4**

**Bicheng Fang**

**Nov 15th, 2016**

This program has simulated a mini file system by creating a file on the disk and using that file as a virtual disk. The maximum length of this virtual disk is 128 blocks and each block has 16 bytes. As this is a simple task, there’s no need to implement the file direction (i.e. all files are stored in the root directory). The maximum number of files is 8, and the maximum number of files being opened at one time is 4.

**Design:**

|  |  |
| --- | --- |
| Super block | 1 block |
| Directory entry | 8 blocks; |
| Inode bitmap: | 1 block; |
| Data bitmap: | 1 block; |
| Inode datamap: | 32 blocks; |
| Data region: | 64 blocks; |

The directory entry is used to record each file’s statues, file name, file length, inode block number. The inode bitmap and data bitmap are used to determine which block or blocks are used for inode, and data. In this project, I also maintain an open-file-table to record which files are being opened. A file can be read or write only if it has been opened.

To implement such file system, the system must know all meta data to access to the stored data. Once a disk has been mounted (i.e. initialized), the system will initialize the super block, directory entry, inode bitmap, data bitmap and inode datamap. After that, files stored on disk can be accessed easily.

After modification of this system, there’s no need to write back all the updated data right away. In this implementation, a disk should be write back only when the file system is dismounted.

**Implementation:**

This program has 12 sub-functions: make\_fs, mount\_fs , dismount\_fs, fs\_open, fs\_close, fs\_create, fs\_delete, fs\_read, fs\_write, fs\_get\_filesize, fs\_lseek, and fs\_truncate.

The first three functions programmed the operation of the file system. Make\_fs function creates a file and initialized the content by all 0s.

In the mount\_fs function, the metablocks of the virtual disk are read into the memory. And I stored that information on a global char array.

In the dismount\_fs function, the metablock is written back to virtual disk by the block\_write function in disk.c

The rest nine functions are used to describe the operation on the file on the disk. The routine operation contains opening file, closing file, deleting file, and modifying file. First, we create a new file by calling fs\_create, it will update the DIR table, and the FAT table. When this file is opend, fs\_open is called and the function will have updated the OFT table in the memory. The offset is set to zero, when the file is newly created. The fs\_write function will modify the content of the file. And the distribution of the block required by the file follows the FIFO principle. The routine is that at first it will look into the OFT table, find the file descriptor and fetch the allocation of the first block of that file according to the file descriptor. Secondly, it will find the free block for this file according to the FAT Table. The rest of these functions are very similar, and the most important task of these functions is to maintain and update the metadata of the file system.