# File System

Written by: Aman Priyadarshi

Writing a file system is challenging task because it requires a memory management and speed efficiency of code. All file system has some basic attributes like Read/Write/Delete/Copy etc.

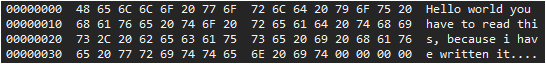
Many more advanced file system are designed which has different features according to their versions like NTFS support a high level of data security and compression…Ext 4 is fastest and very cool file system, they also have file recovery option…Every file system has a basic structure and data arrangement algorithm because the file system is like a test of programmers brain (according to me.)

Almost every file system has few things common i.e. the file table (literally saying…) The FAT 32 allocate few blocks in starting which saves the general information like the volume label, number of fat available, media of device (i.e . floppy/cd-rom/hdd etc.) serial number. I’m not going to discuss whole file system structure here, for that I provided the links.

So, Why the few file systems are good enough while other are poor. The reason behind it is the coding style/using algorithm and encryption type. The Ext use [binary tree](http://en.wikipedia.org/wiki/Binary_tree) for directory listening while NTFS has MFT (Master file table record)…it is not one in whole hard disk…because then it becomes file system too much slow. Now there are few restrictions while making a file system which makes it a challenging task for programmers to design a file system… If you take a look at FAT 12 which was developed by bill gates (I don’t like him) is not have a good algorithm (not saying because I don’t like him) the file system use search method for a finding a file in a small portion of disk…Now why this is stupid algorithm, first is maximum size…it can hold maximum 12GB, second it makes file system very slow and third it have less attributes file data loss, limited size of file and name length etc. Now why this is a famous file system and re implemented as FAT 16 and FAT 32…first is easy way to implement. Second, at that time speed is not much too much taken in consideration. And last but not least it was the success full file system of that time…(I am thinking why I was not at that time, well if I were at that time it maybe that I even don’t know what the hell is computer xD)

Come on topic once again, Hard disk can read and write in certain block size limit… X) now what is this? So every hard disk promises a programmer that I always give you array of bytes from a offset multiple of that length….lets us understood with an example

Suppose this is a hard disk data clone



Let the fixed array size is 8, so when we ask our friend (i.e. hard disk xD) to give us data…then he (xD) will ask from which block number…for example you said “2” then your friend will return {68, 61, 76, 65, 20, 74, 6F, 20} (second row first column)…what the hell is this?

So the block is fixed array in hard disk…so you can say a hard disk is a collection of many blocks. This hard disk has 8 blocks. Starting number from 0, 1, 2….7

YOU CAN’t read alone byte from any block…you have to read whole block at one…you can also read more than blocks at one time. Now why our friend don’t give us a single byte from given block…it is because the hard disk reading is done via IO/Port so which can read only in a collection of WORD at a time. This constant for any IDE/SATA hard disk is 512 bytes. While most of atapi (cd-rom) have this value 1024 bytes.

Okay I told you all basics of writing a file system rest of work is your programming and brain…Good Luck!!

Here are few links that may help you =)

<http://wiki.osdev.org/FAT>

<http://wiki.osdev.org/NTFS>

<http://wiki.osdev.org/Ext2>

<https://www.pjrc.com/tech/8051/ide/fat32.html>

<http://staff.washington.edu/dittrich/misc/fatgen103.pdf>

<http://discutils.codeplex.com/>

I also wrote a file system about 1 year ago, that was also cool…but not too much

<https://www.youtube.com/watch?v=D6h5iUkfJg0>

<https://mega.co.nz/#!RIdCTLrY!BBF190MMjU-qv3IzQbdVugBlUQdEqO0E4M7ePlFKzWE>

And the last but not least.

<https://www.google.com/>

Good Luck =)