

# **PROJECT REPORT**

**TITLE:** ANALYSIS OF FILE SYSTEM

## **TEAM MEMBERS:**

<b>NAME:</b>	<b>REG. NO.</b>
VIDIT	17BIT0246
MUKUL SINGH LOOHAHCH	17BIT0245

## **What is File System?**

File system controls how data is stored and retrieved. Without a file system, information placed in a storage medium would be one large body of data with no way to tell where one piece of information stops and the next begins. By separating the data into pieces and giving each piece a name, the information is easily isolated and identified.

There are many different kinds of file systems. Each one has different structure and logic, properties of speed, flexibility, security, size and more. Some file systems have been designed to be used for specific applications.

The Microsoft Windows working frameworks have constantly bolstered, and still do bolster, different variants of the FAT (File Allocation Table) record framework.

Notwithstanding FAT, all Microsoft Windows working frameworks since Windows NT support a more up to date record framework called NTFS (New Technology File System).

Every advanced variant of Windows likewise support exFAT, a file system intended for Flash drives.

## **What Is FAT32 File System?**

An acronym for **File Allocation Table**, it's the oldest and the most experienced file system in the history of computing. The story started in 1977 with the original 8-bit FAT file system intended as an antecedent to Microsoft's Standalone Disk Basic-80 which was launched for Intel's 8080-based NCR 7200 in 1977/1978 — a data entry terminal with 8-inch floppy disks. It was coded by Microsoft's first salaried employee Marc McDonald following discussions with Microsoft's co-founder Bill Gates.

Further utilization of the FAT File System, or FAT Structure, as it was called earlier, was done in Microsoft's 8080/Z80 platform-based operating system MDOS/MIDAS written by Marc McDonald.

### **FAT32: Limits and compatibility**

In the subsequent years, the FAT file system advanced to FAT12, FAT16, and finally, FAT32 that has been synonymous with the word file system when we have to deal external storage media like removable drives.

FAT32 surmounted the limited volume size offered by the FAT16 file system. The 32-bit File Allocation Table was released in August 1995, with the launch of the Windows 95 operating system. FAT32 allows you to store files of size up to 4GB and the maximum disk size can go up to 16TB.

So, the fatty file system can't be used to install heavy apps or store large files, that's why the modern Windows uses a new file system known as NTFS, for which we don't have to worry about the file size and disk size limits.

Almost all versions of Windows, Mac, and Linux are compatible with FAT32 filesystem.

## When to choose FAT32?

The FAT32 file system is perfect for storage devices like flash drives but you'll have to make sure you don't have an individual file larger than 4GB. It has been widely implemented beyond the realm of computers, like gaming consoles, HDTVs, DVD & Blu-Ray players, and practically any device with a USB port.

**Compatibility:** Works with all versions of Windows, but read-only with Mac by default, and may be read-only by default with some Linux distributions. Other devices—with the exception of Microsoft's Xbox One—probably won't support NTFS.

**Limits:** No realistic file-size or partition size limits.

**Ideal Use:** Use it for your Windows system drive and other internal drives that will just be used with Windows.

## How FAT32 WORKS?

A FAT file system is composed of four different sections:

The Reserved sectors, located at the very beginning. The first reserved sector (sector 0) is the Boot Sector (aka Volume Boot Record (VBR)). It includes an area called the BIOS Parameter Block (with some basic file system information, in particular its type, and pointers to the location of the other sections) and usually contains the operating system's boot loader code. Important information from the Boot Sector is accessible through an operating system

structure called the Drive Parameter Block (DPB) in DOS and OS/2. The total count of reserved sectors is indicated by a field inside the Boot Sector. For FAT32 file systems, the reserved sectors include a File System Information Sector at sector 1 and a Backup Boot Sector at sector 6.

### **The FAT Region.**

This typically contains two copies (may vary) of the File Allocation Table for the sake of redundancy checking, although rarely used, even by disk repair utilities. These are maps of the Data Region, indicating which clusters are used by files and directories. In FAT12 and FAT16 they immediately follow the reserved sectors. Typically the extra copies are kept in tight synchronization on writes, and on reads they are only used when errors occur in the first FAT. In FAT32, it is possible to switch from the default behaviour and select a single FAT out of the available ones to be used for diagnosis purposes.

### **The Root Directory Region.**

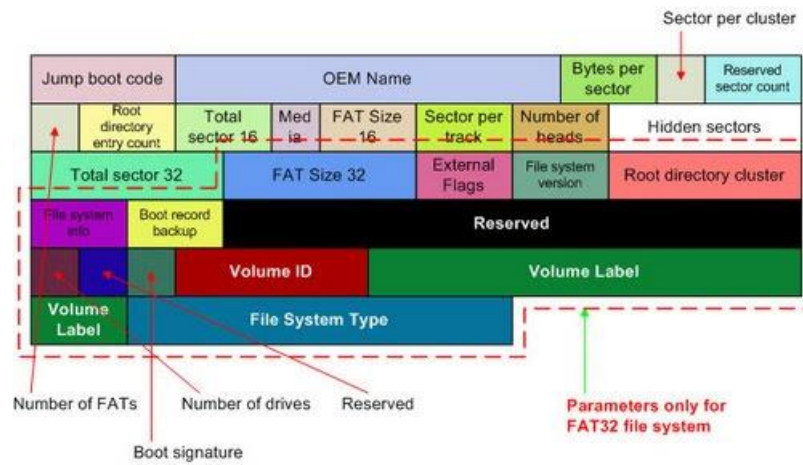
This is a Directory Table that stores information about the files and directories located in the root directory. It is only used with FAT12 and FAT16, and imposes on the root directory a fixed maximum size which is pre-allocated at creation of this volume. FAT32 stores the root directory in the Data Region, along with files and other directories, allowing it to grow without such a constraint. Thus, for FAT32, the Data Region starts here.

### **The Data Region.**

This is where the actual file and directory data is stored and takes up most of the partition. Traditionally, the unused parts of the data region are initialized with a filler value of 0xF6 during format on IBM compatible machines, but also used on the Atari Portfolio. 8-inch CP/M floppies typically came pre-formatted with a value of 0xE5;<sup>[9]</sup> by way of Digital Research this value was also used on Atari ST formatted floppies. Some modern formatters wipe hard disks with a value of 0x00, whereas a value of 0xFF is used on flash disks to reduce wear. The latter value is typically also used on ROM disks. (Some advanced formatting tools allow to configure the format filler byte.)

The size of files and subdirectories can be increased arbitrarily (as long as there are free clusters) by simply adding more links to the file's chain in the FAT. Note however, that files are allocated in units of clusters, so if a 1 KB file resides in a 32 KB cluster, 31 KB are wasted.

FAT32 typically commences the Root Directory Table in cluster number 2: the first cluster of the Data Region.



The items of FAT32 boot record

by *iprinceps*, <iprinceps@gmail.com>

## Performance:

Our main goal is to analyse the variation in performance of the disk and the access times before and after executing the defragmentation.

## Implementation:

```
1. Create a file
2. Delete a file
3. Defragmentation
4. Display the Application Files
5. System File Creation
6. Display System Files
7. Search for a File
8. Deletion of files older than 1 month
9. Exit
Enter your choice:
```

Enter your choice:1

Enter file Name:vidit  
Enter File Size:256  
Enter File type:pdf

\*\*\*\*\* Displaying Application File in the System(FAT 32) \*\*\*\*\*

NAME	S_ADDR	SIZE	TYPE	TIMESTAMP
vidit	0	256	pdf	Sat Apr 06 08:47:13 2019

File Added..

Enter your choice:2

Enter file name which you want to delete:abc

\*\*\*\*\* Displaying Application File in the System(FAT 32) \*\*\*\*\*

NAME	S_ADDR	SIZE	TYPE	TIMESTAMP
abc1	0	256	txt	Sat Apr 06 08:48:43 2019
vidit	768	256	docx	Sat Apr 06 08:48:07 2019
vidit	1024	256	pdf	Sat Apr 06 08:47:59 2019

File deleted..

Enter your choice:3

Defragmentation in Progress.. Reseting all the addresses!

\*\*\*\*\* Displaying Application File in the System(FAT 32) \*\*\*\*\*

NAME	S_ADDR	SIZE	TYPE	TIMESTAMP
abc1	0	256	txt	Sat Apr 06 08:48:43 2019
vidit	256	256	docx	Sat Apr 06 08:48:07 2019
vidit	512	256	pdf	Sat Apr 06 08:47:59 2019

Defragmentation Successfully completed.

Enter your choice:4

Displaying the Application Files:

\*\*\*\*\* Displaying Application File in the System(FAT 32) \*\*\*\*\*

NAME	S_ADDR	SIZE	TYPE	TIMESTAMP
abc1	0	256	txt	Sat Apr 06 08:48:43 2019
vidit	256	256	docx	Sat Apr 06 08:48:07 2019
vidit	512	256	pdf	Sat Apr 06 08:47:59 2019

Enter your choice:5

Enter file Name:abc

Enter File Size:256

Enter File type:bat

File Added in priority list..

Enter your choice:6

Do you Really want to see the System Files[y/n]:y

Enter the Password if you want to see the System files:admin

\*\*\*\*\* Displaying System Files(FAT 32) \*\*\*\*\*

NAME	S_ADDR	SIZE	TYPE
abc	0	256	bat

Enter your choice:7

Enter the File Name: vidit

Searching in progress...

vidit	256	256	docx	Sat Apr 06 08:48:07 2019
vidit	512	256	pdf	Sat Apr 06 08:47:59 2019

Searching is progress..

All files are new..

\*\*\*\*\* Displaying Application File in the System(FAT 32) \*\*\*\*\*

NAME	S_ADDR	SIZE	TYPE	TIMESTAMP
abc1	0	256	txt	Sat Apr 06 08:48:43 2019
vidit	256	256	docx	Sat Apr 06 08:48:07 2019
vidit	512	256	pdf	Sat Apr 06 08:47:59 2019

Enter your choice:9

Good Bye):

Process returned 0 (0x0) execution time : 423.787 s

Press any key to continue.

## What Is NTFS File System?

Another Microsoft proprietary file system called NTFS (**New Technology File System**) was **introduced in 1993** with their Windows NT 3.1 operating system coming into existence.

NTFS file system offers inexhaustible file size limits. As of now, it would be next to impossible for us to even reach somewhere near the boundary. The development of NTFS file system started in the mid-1980s as the result of an association between Microsoft and IBM to develop a new age operating system with better performance in terms of graphics.

However, their friendship didn't last long and the two parted their ways, consequently developing their own version of the new file system. In 1989, IBM made HPFS which was used in the OS/2 operating system while the partnership was still going on. Microsoft launched NTFS v1.0 and released it with Windows NT 3.1 in 1993.

### NTFS: Limits and features

The NTFS file system offers a **theoretical file size of 16 EB – 1 KB**, which is **18,446,744,073,709,550,592 bytes**. Well, your files aren't that much huge, I guess. Its development team included Tom Miller, Garry Kimura, Brian Andrew, and David Goebel.



NTFS v3.1 was launched with Microsoft's Windows XP and it hasn't been changed much since then, although, numerous additions like partition shrinking, self-healing, and NTFS Symbolic links have been added. Also, the implemented capacity of NTFS file system is only 256 TB out the whopping 16 EB – 1 KB which was done with the launch of Windows 8, for which Microsoft

Also, the implemented capacity of NTFS file system is only 256 TB out the whopping 16 EB – 1KB which was done with the launch of Windows 8.

Other notable features include reparse points, sparse file support, disk usage quotas, distributed link tracking, and file-level encryption. The NTFS file system supports backward compatibility with the previous versions.

It's a journaling file system which proves to be an important aspect when it comes to reviving a corrupt file system. It maintains a Journal, a data structure which keeps track of any potential modifications to the file system and is used to recover the file system.

The NTFS file system is supported by Windows XP and later versions. Apple's Mac OSX provides read-only support for an NTFS-formatted drive and only a few Linux variants are able to provide write support for NTFS.

**Compatibility:** Works with all versions of Windows, Mac, Linux, game consoles, and practically anything with a USB port.

**Limits:** 4 GB maximum file size, 8 TB maximum partition size.

**Ideal Use:** Use it on removable drives where you need maximum compatibility with the widest range of devices, assuming you don't have any files 4 GB or larger in size.

## How NTFS Works?

When a hard disk is formatted, it is divided into partitions of the total physical hard disk space. Within each partition, the operating system keeps track of all the files that are stored by that operating system. Each file is actually stored on the hard disk in one or more clusters of a predefined uniform size. Using NTFS, the sizes of clusters range from 512 bytes to 64 kilobytes. Windows NT provides a recommended default cluster size for any given drive size. (Note that clusters are indivisible. Even the smallest file takes up one cluster and a 4.1 KB file takes up two clusters or 8 KB on a 4 KB cluster system)

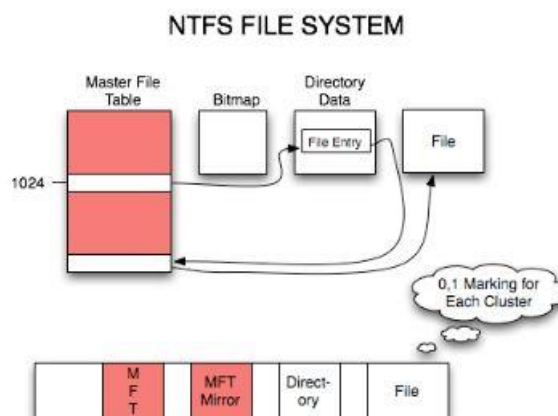
In general, using NTFS, the larger the hard disk the larger the default cluster size, since it's assumed that a system user will prefer to increase performance (fewer disk accesses) at the expense of some amount of space inefficiency.

Everything within NTFS is considered a file, and that applies to directories as well. Each directory has an entry in the Master File Table, which serves as the main repository of information for the directory. The MFT record for the directory contains the following information and NTFS attributes:

- **Header (H):** This is a set of low-level management data used by NTFS to manage the directory. It includes sequence numbers used internally by NTFS and pointers to the directory's attributes and free space within the record. (Note that the header is part of the MFT record but not an attribute.)
- **Standard Information Attribute (SI):** This attribute contains "standard" information stored for all files and directories. This includes fundamental properties such as date/time-stamps for when the directory was created, modified and accessed. It also contains the "standard" attributes usually associated with a file (such as whether the file is read-only, hidden, and so on.)
- **File Name Attribute (FN):** This attribute stores the name associated with the directory. Note that a directory can have multiple file name

attributes, to allow the storage of the "regular" name of the file, along with an MS-DOS short filename alias and also POSIX-like hard links from multiple directories.

- **Index Root Attribute:** This attribute contains the actual index of files contained within the directory, or part of the index if it is large. If the directory is small, the entire index will fit within this attribute in the MFT; if it is too large, some of the information is here and the rest is stored in external index buffer attributes, as described below.
- **Index Allocation Attribute:** If a directory index is too large to fit in the index root attribute, the MFT record for the directory will contain an index allocation attribute, which contains pointers to index buffer entries containing the rest of the directory's index information.
- **Security Descriptor (SD) Attribute:** This attribute contains security information that controls access to the directory and its contents. The directory's Access Control Lists (ACLs) and related data are stored here.



## What Is exFAT File System?

The **exFAT (Extended FAT)** is another **Microsoft proprietary file system** which finds its use in ball games where the FAT32 feels out of breath. Most of the modern digital cameras use exFAT. High capacity SDXC memory cards are now

pre-formatted with the exFAT file system, as it is lighter in contrast to NTFS and supports file of sizes, more than 4GB. So, if you have an exFAT SD card, you shall face no issues while copying full-length HD movies on it which is not the case with FAT32.

### **exFAT: Limits and features**

High capacity SDXC memory cards are now pre-formatted with the exFAT, as it is lightweight in contrast to NTFS and **supports file size more of than 4 GB**. So, if you have an exFAT SD card you will have no issues while copying full-length HD movies on it which is not the case with FAT32.

Currently, Microsoft holds the US Patent 8321439 for Quick File Name Lookup using Name hash. It is a method which escalates file search speed. They've not released a complete specification set for the exFAT file system, for which the vendors have to obtain a restricted license from Microsoft.

Most of the countries recognize the US Patent Law and thus, any implementation of the exFAT file system is not possible for the vendors which intend it to be a part of an open-source operating system or a commercial software.

A FUSE-based version of the exFAT file system, named exfat-fuse, provides read/write operations for many Linux distributions including FreeBSD. A kernel-based implementation made by Samsung was accidentally leaked on GitHub but was later published officially.

Launched in 2006, exFAT file system has the same 16 EB file size limit as NTFS, but it is way lighter as it doesn't contain many of the extra features that the latter has. Talking about the compatibility, full read and write support is provided by Mac, Android, and Windows operating systems. But for Linux distributions, the appropriate software facilitates the support.

**Compatibility:** Works with all versions of Windows and modern versions of Mac OS X, but requires additional software on Linux. More devices support exFAT than support NTFS, but some—particularly older ones—may only support FAT32.

**Limits:** No realistic file-size or partition-size limits.

**Ideal Use:** Use it when you need bigger file size and partition limits than FAT32 offers and when you need more compatibility than NTFS offers. Assuming that every device you want to use the drive with supports exFAT, you should format your device with exFAT instead of FAT32.

## NTFS vs FAT32 vs exFAT: Comparison

	FAT32	NTFS	exFAT
Compatibility	Windows XP and later versions, Mac OSX 10.6.5 and above, Linux (using FUSE), Android	Windows, Mac (Read-only), Linux (read-only for some distributions), Microsoft Xbox One.	Windows XP and later versions, Mac OSX 10.6.5 and above, Linux (using FUSE), Android
Pros	Cross-platform compatibility, lightweight.	Inexhaustible limits for file and partition size	Behaves as a fusion between FAT32 and NTFS by providing practically unlimited file size and partition size.
Cons	Limited file size (up to 4GB) and partition size (up to 16TB)	Limited cross-platform compatibility	Microsoft restricts its usage by license obligations.
Usage	Removable storage devices.	Best for internal hard drives. Use it for Windows system drive and	Use with external hard drives or flash drives if you want to transfer files larger than 4 GB.

		any other system partition which may be used to install software.	Always prefer exFAT over FAT32, if the operating system supports it.
--	--	---	--