ANALYSIS OF THE FILESYSTEM IN MAC OS X (APFS- APPLE FILE SYSTEM)

by

GUGAN S KATHIRESAN 18BLC1089
ANIRUDH MUTHUSWAMY 18BLC1135
TARUN RAHUL 18BLC1160

A project report submitted to

Dr. MAHESWARI R

SCHOOL OF ELECTRONICS ENGINEERING

in partial fulfilment of the requirements for the course of

Operating Systems(CSE2005)

in

B.Tech. ELECTRONICS AND COMPUTER SCIENCE ENGINEERING

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

VIT CHENNAI

Vandalur – Kelambakkam Road

Chennai – 600127

MAY 2020

BONAFIDE CERTIFICATE

Certified that this project report entitled "ANALYSIS OF THE FILESYSTEM IN MAC OS X (APFS- APPLE FILE SYSTEM)" is a bonafide work of ANIRUDH MUTHUSWAMY - 18BLC1135, GUGAN KATHIRESAN – 18BLC1089, TARUN RAHUL - 18BLC1160 who carried out the Project work under my supervision and guidance for **Operating Systems(CSE2005)**

Dr. MAHESWARI R

Associate Professor Senior

School of Computer Science Engineering (SCSE),

VIT University, Chennai

Chennai – 600 127.

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, Dr. MAHESWARI R , School of computer science engineering, for her consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

TABLE OF CONTENTS

ABSTRACT

In this project we look into what a file system is and what file systems Apple's MAC OS X has used over the past years and understand how with every new file system implemented by Apple is different from the one that came before it. We then understand how APFS is special and what features make it stand out as the modern file system meant to optimize current generation hardware in the form of flash storage and solid state drives. We then understand how APFS is different from the other file systems found on other platforms such as Windows and Linux.

1. INTRODUCTION TO FILE SYSTEMS

1.1 What is a file system?

A filesystem is the methods and data structure that the OS utilizes to know where the files are present, how the data must be accessed and how it must be managed. They work by creating an index for all the data contained in a storage device.

1.2 History of File systems on Mac OS X

- **HFS**

  HFS or Hierarchical File System was the first file system used by Apple in MAC OS X. It was the main filesystem format utilized in mac OS until MAC OS 8.1. After which

- **HFS+ or HFS Plus**

  The file system used by Apple to overcome the flaws of HFS by improving reliability in MAC OS X

- **APFS**

The file system currently being used by Apple which is optimized for solid state drives and flash storage

1.3 How is APFS different from file systems used in Windows and Linux

Windows uses ExFAT which is the best cross platform option which can work on both windows and macOS.
Linux uses ext2,3,4 which is proprietary to linux the same way APFS is proprietary to macOS.

2. What is a file system?

A file system is what controls the manner in which data is stored and retrieved in a storage medium. File systems are made up of files which are divided into groups which are called directories. Directories can either hold files or more directories within them.

File systems have existed for a long time and have seen multiple revisions since its first inception to improve its efficiency and reliability.

As a whole we can say that a file system is a structured representation of data and a set of metadata describing the data.

As the user keeps performing write/delete operations in the file system, it causes defragmentation. Thus, files that were once stored as units are then divided into fragments.

Under normal circumstances the file system works in blocks instead of sectors. These file system blocks are collections of sectors that result in optimization of storage addressing.

The file systems we use nowadays utilize block sizes which vary from 1 to 128 sectors.

Out of which the files are stored at the beginning of the block and use the entire block.

3. File systems in macOS

macOS uses two file systems, which are HFS+ and APFS.

HFS+ is just an extension to the previously existing HFS file system or

Hierarchical File system which was intended to be used on older Macintosh computers, whereas APFS is a more recently introduced filesystem which is more optimized for the current Solid state drive technology.

Apple uses different type of file systems throughout its life. First it used HFS (Hierarchical file system), then HFS plus which was an improvement of HFS in terms of reliability. HFS plus was introduced in MAC OS X whereas HFS was introduced in system 2.X (1985)

The newest version of this File system is APFS which is made with Solid state drives and encryption in mind.

This file system in MAC OS manages how data is stored and retrieved on a storage medium efficiently, as all apple devices ranging from the apple watch to the Mac pro uses the same file system. Hence it’s scalability is much wider compared to traditional file systems used in windows and Linux.

4. APFS and its key features

APFS uses **64-bit inode numbers.**

Inode number is another term for an index number. Inode is a number which is assigned to files and directories when it is being made. Each number is one of a kind and cant be replicated.

The following information is stored in inode:

- **File type:** regular file, directory, pipe etc.
- **Permissions to that file:** read, write, execute
- **User ID:** owner of file
- **Link count:** Count of hard link with respect to the inode
- **Other metadata** about the file
- **Size of file:** The amount of storage a file utilizes
- **Time stamp**: Used to know when and how the modifications were made
- **Group ID:** group owner
- **Attributes:** immutable' for example
- **Access control list:** permissions for special users/groups
- **Link to location of file**

<u>The benefits of APFS are:</u>

1. Increased speed
2. Save power
3. Save space
4. Fight against crashes
5. Help backup data more efficiently
6. Theoretically be more secure

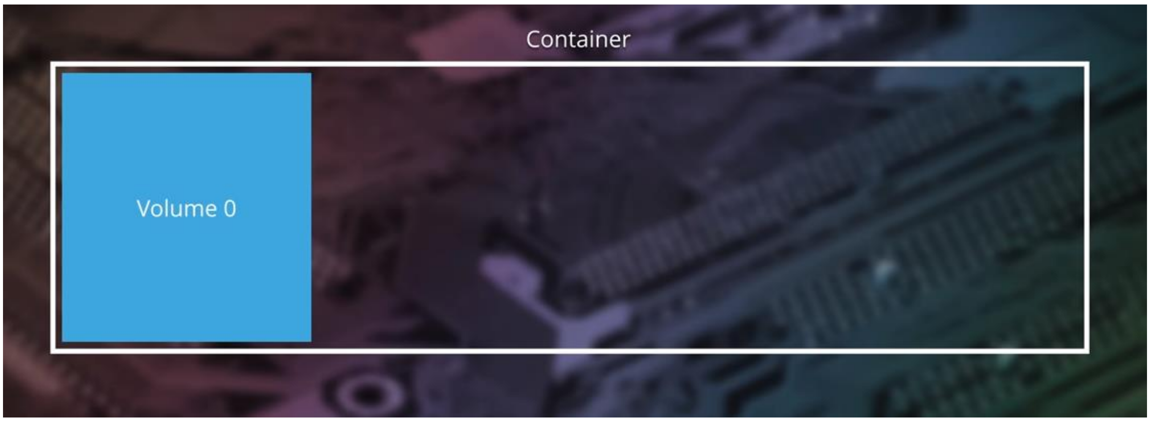
How do these benefits work?

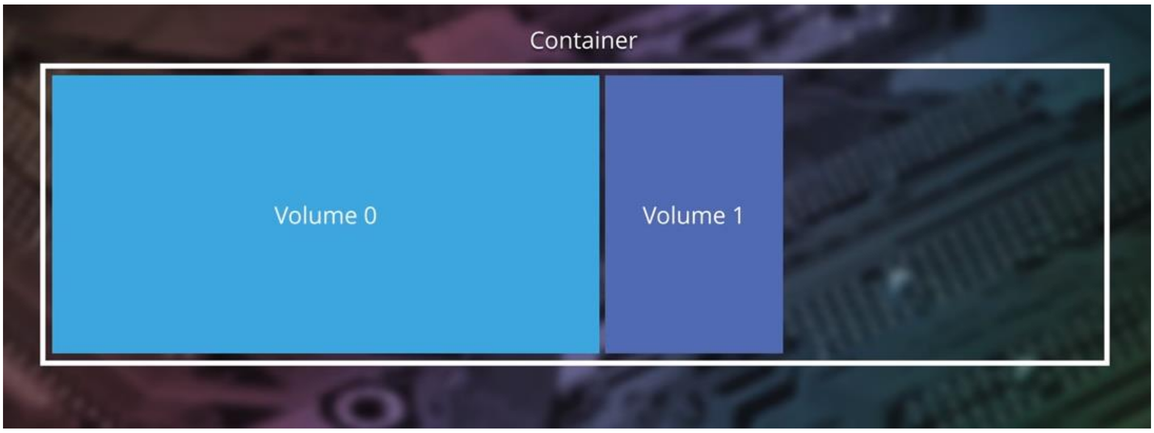
The key features of APFS:

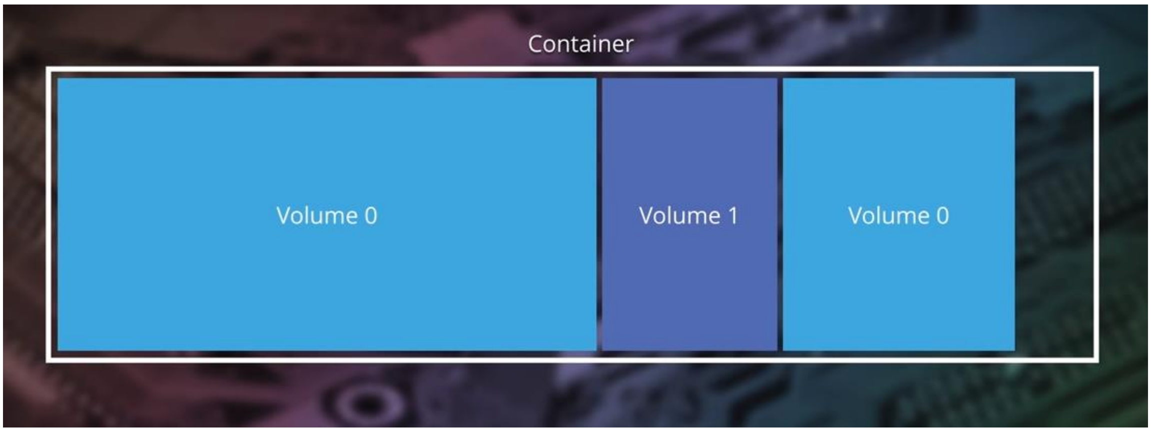
1. Space sharing
2. Cloning
3. Snapshots
4. Copy on Write
5. Atomic Safe-Save

SPACE SHARING:

When you partition a disk typically you have pre-allocated volumes that are not dynamic. That is the volumes have set or fixed amounts of space for the data. However in APFS when you partition a disk you don’t make a separate volume, you make a container and there’s volumes inside the container that is dynamic. These volumes which are within the container share a free space, and the user has control over how the free space can be used. In the latest MAC OS, when volumes are created within a container, the user can set a reserve size and a quota size for each volume.







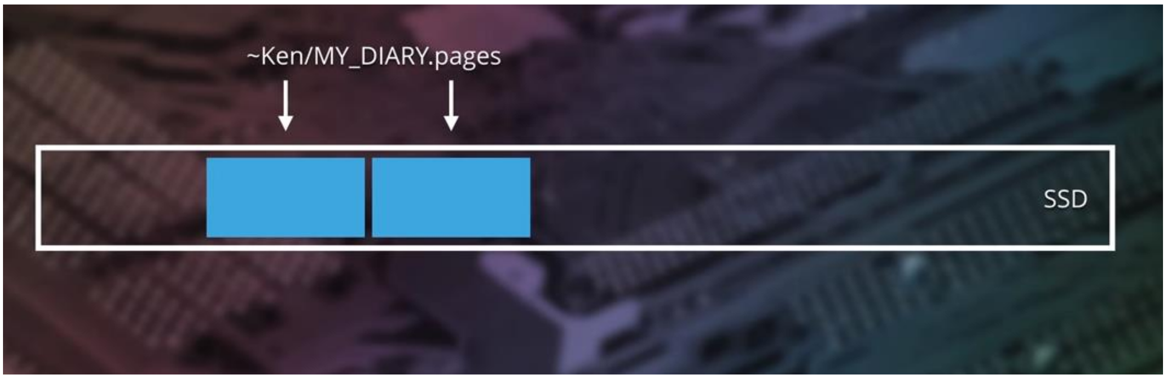
As you can see, the volumes can grow or shrink within the container in Apple file systems, i.e. the partitioning of disks is dynamic.

COPY ON WRITE (COW)

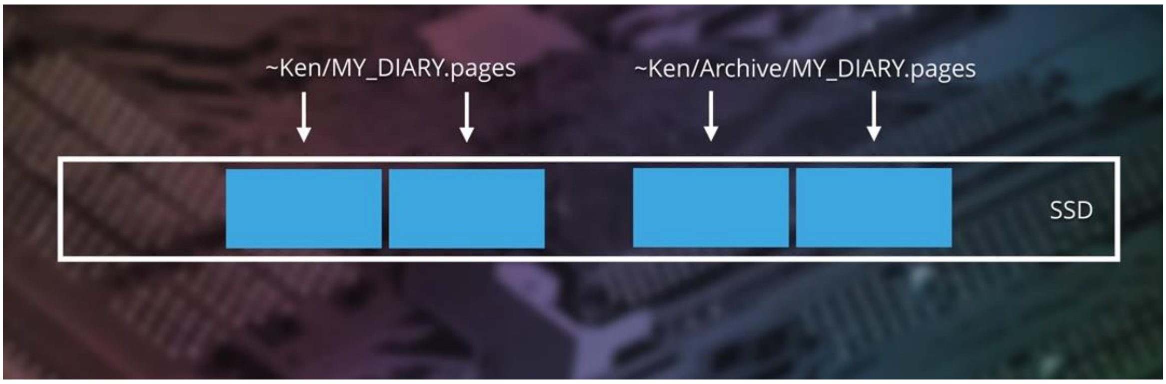
Copy on Write is an optimization strategy. In APFS, this can help save space, prevent crashes, and boosts speed.

Basically, Copies of files are only made when changes need to be written. This concept might not be new entirely, but the way Apple implements it is completely different.

Let's say we have two blocks of data on the solid-state drive and for archival purposes we need to make a duplicate of these files.

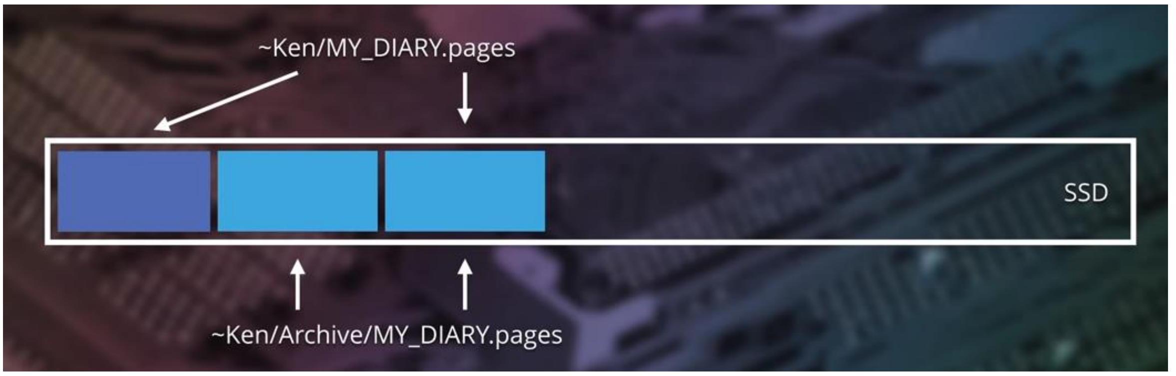


So now, two files are using four blocks of data on the SSD



This is the traditional way of doing it. But by using Copy On Write, there's only 2 blocks as opposed to 4, and we can save the extra computing power or time to read and write the data. Plus we saved extra hard drive space by just referring to those two blocks of data.

If any changes are to be made, then the changes alone are written out into a separate block of data called a delta folder. It doesn't necessarily have to duplicate the file, only the additional changes are copied into the delta folder.



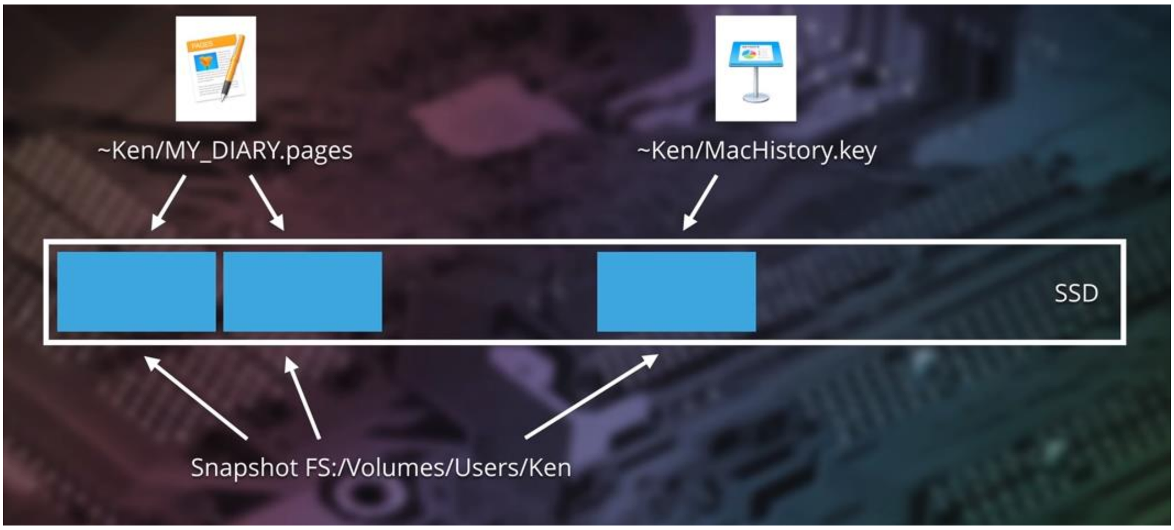
CLONING

A cloned file can be copied instantly and can take up almost no additional space. A small amount of space is used for the metadata of the cloned file because the system needs to understand where to point to the block of data.

Cloning copies the references to the data and not the actual data.

SNAPSHOTS

Using the copy on write method APFS creates snapshots. These are read-only instances of the file system. Suppose we have 2 files that take up 3 blocks of data in the SSD.



So by taking a snapshot, only a read-only instances of these blocks of data is saved in the OS. Even If changes are written to the files, the snapshot is still preserved in the same drive and can be retrieved.

If a file is deleted, the file system won't reclaim blocks that are backed up in the snapshot.

ATOMIC SAFE-SAVE

Today, regular files save into a temporary directory during a safe save. That is the data is stored on to a temporary folder and when the data is verified, the old file is removed and the new file is renamed Automatically.

An Atomic Operation means that it happens instantaneously to the end user.

Whereas in document bundles, which are basically folders which acts like a single file. They contain assets for the file within itself.

In HFS+ when you perform a save on these document bundles, it isn't atomic. This can cause a risk as data can be lost during a save operation.

This is due to a POSIX (Portable Operating System Interface) limitation.

APFS bypasses this POSIX limitation and atomically performs the swap. So, if there was a crash or a power loss, the data wouldn't be gone as the safe save operation would happen instantaneously in APFS.

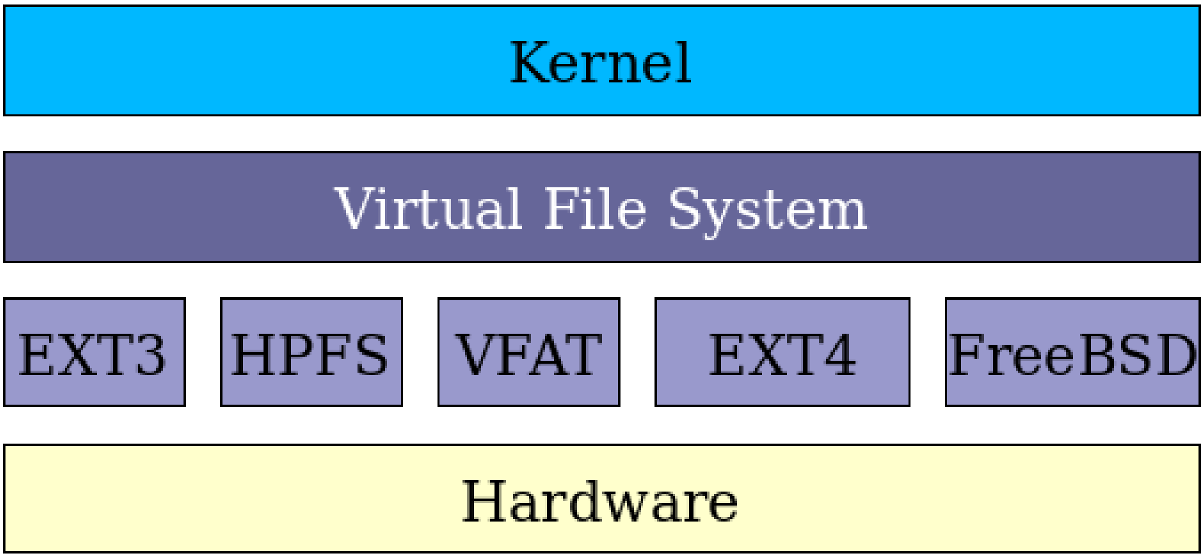
This is done using a new system called renamex_np (np-stands for NON-POSIX).

OTHER KEY FEATURES OF APFS COMPARE TO HFS+

1. Supports 9 quintillion files as opposed to the 4 billion limit.
2. 1 nanosecond time stamp.
3. Multi-key encryption.
4. Fast directory sizing.
5. Sparse files.
6. Intelligent HDD defragmenter.
7. Boatload of new APIs.

5. How APFS is different from other file systems

Difference between APFS and EXT3



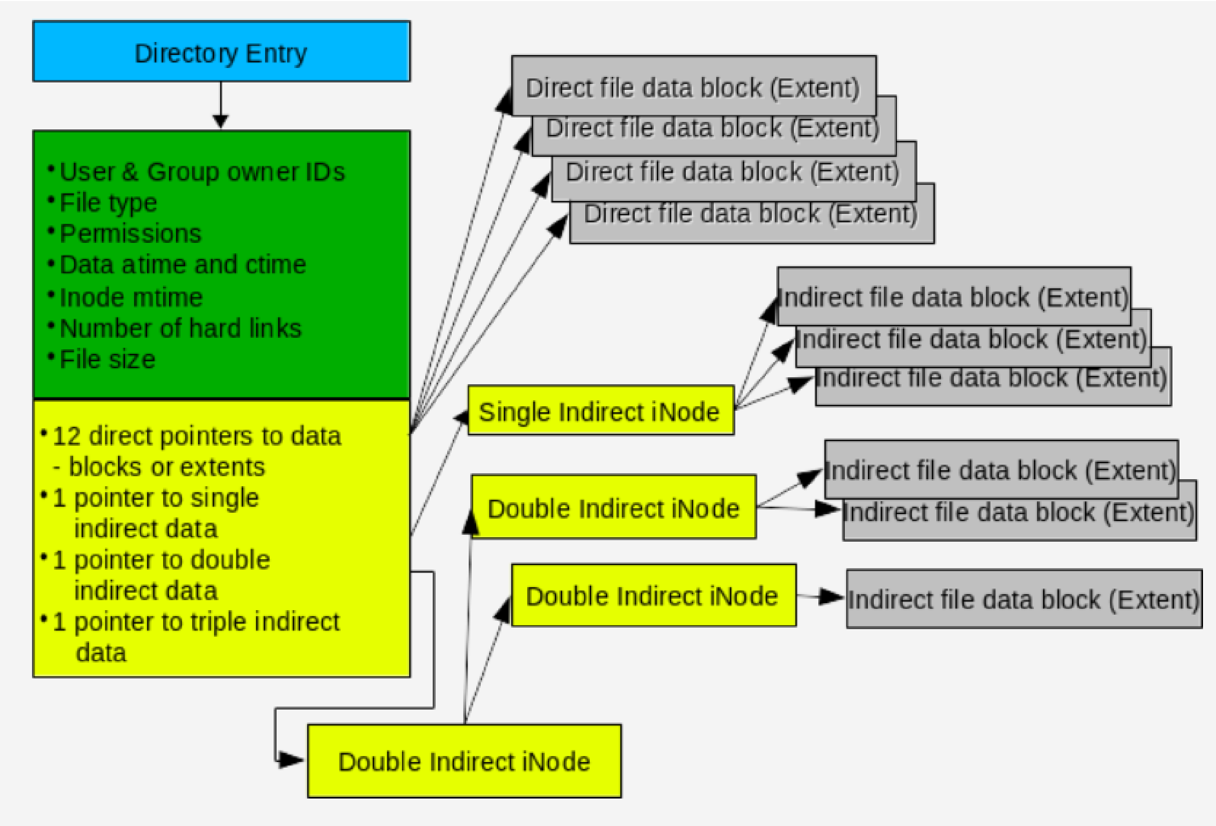
Linux on the other hand uses EXT filesystem. This filesystem has been updated through the years from EXT2 to EXT4.

The EXT4 filesystem mainly focuses on improvements In the form of performance , reliability and capacity.

Even though EXT and APFS use the same inode data structures used in UNIX to store the file metadata, the design of this filesystem is completely different from APFS as EXT4 is not readily supported in Windows or mac OS devices.

An extent is described by its starting and ending place on the hard drive. This makes it possible to describe very long, physically contiguous files in a single inode pointer entry.

This diagram is the directory and inode for a single file which, in this case, may be highly fragmented. The EXT filesystems work in the background to prevent fragmentation, so it is very unlikely you will ever see a file with this many indirect data blocks or extents.



How is APFS different from FAT

Windows on the other hand uses FAT as its file system.

FAT file system works by creating a file allocation table which is why it is called FAT. This table is present at the top of the volume to ensure the volume is safe and it creates a duplicate of this table to ensure that there is always a backup in case one of them get affected. This file is always in a static location to ensure there is easy access to the file.

When a disk uses the FAT file system , the data is allocated in clusters similar to the blocks found in APFS. The size of these clusters depends on the size of the volume. FAT however requires all the updates to be registered on the table making it time consuming and not as efficient as APFS.

6. Conclusion

After performing this in-depth analysis we can see that the APFS is unique in the way that it is proprietary to macOS and it is mainly made for the Solid state drives and is hence well optimized for the current generation hardware. It is the only file system which supports scalability to such a great extent where it can be used on a device such as the apple watch all the way up to the iMac which goes to show the efficiency of the file system. It also keeps a backup of the data in the form of a snapshot which makes it incredibly reliable for users in the event of a catastrophic failure.