

System forensic

Credit: [SystemForensics - YouTube](#)

Noted by: [Tsof \(@tsof_relox\) / X](#)

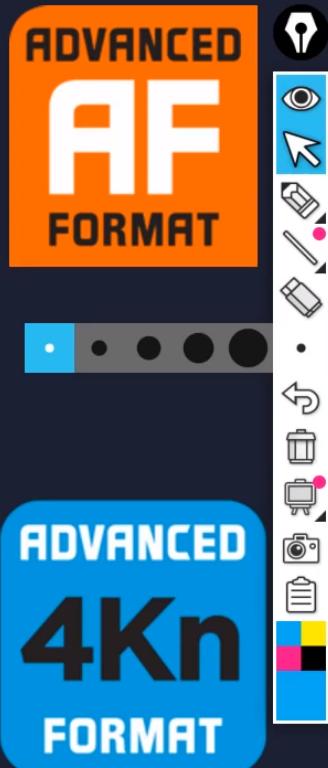
Many thanks to @systemforensics for the great lecture series, myself and many people in the DF community learned a lot from you. With easy-to-understand content, logical presentation and immediately applicable examples. The only problem is the sound, which I don't even remember affecting my learning process.

If you use this note of mine and find it useful, I hope you can take a moment to subscribe and like @systemforensics products. After all he deserves it.

Disk layout - MBR - MPT - GPT

DISK LAYOUT – UNDERSTANDING SECTORS

- Each sector stores a fixed amount of user-accessible data, traditionally 512 bytes **for hard disk drives** (HDDs) Newer HDDs use 4096-byte (4 KiB) sectors, which are known as the Advanced Format (AF). The sector is the minimum storage unit of a hard drive.
- Many host computer hardware and software components assume the hard drive is configured around 512-byte sector boundaries.
- In order to maintain compatibility with legacy computing components, many hard disk drive suppliers support Advanced Format technologies on the recording media coupled with 512-byte conversion firmware, and referred to as Advanced Format 512e, or 512 emulation drives.



DISK LAYOUT – UNDERSTANDING PARTITIONS

- Every sector on a drive is addressed by LBA (Logical Block Address), starting from Sector 0
- Partitioning is logical dividing of the drive storage space (all addressable physical sectors) into several pieces (partitions)
- You can allocate only consecutive sectors into one partition
- The number of possible partitions on a drive will depend on the chosen Partitioning Style



DISK LAYOUT - PARTITION STYLES: MBR

- MBR – stands for Master Boot Record
- MBR style is common for the external USB drives (memory sticks, removable hard drives, SD-cards and Linux systems)
- MBR is located in Sector 0 and contains the Master Partition Table (MPT)
- MBR has a flag 0x55AA at the end of the sector
- Master Partition Table can accommodate up to 4 partition entries - Primary or Extended

DISK LAYOUT - PARTITION STYLES: MPT

- MPT – Master Partition table, 64 bytes long
- Located at the end of the MBR sector, at the offset 446 bytes from the beginning of the sector
- Each partition entry is 16 bytes long: offsets within entry:
 - offset 0 (1st byte) – boot indicator (0x80 – bootable partition, OS system partition)*
 - offset 4 (5th byte) – Partition type descriptor – indicates file system type on the partition*
 - offset 8-11 (9th to 12th bytes) – Starting Sector of the partition*
 - offset 12-15 (13th to 16th bytes) – Partition Size (in sectors)*

File View Mode Help



Evidence Tree

+ \PHYSICALDRIVE2

File List

Name Size Type Date Modified



Hex Value Interpreter

Type	Size	Value
signed integer	1-8	
unsigned integer	1-8	
FILETIME (UTC)	8	
FILETIME (local)	8	
DOS date	2	
DOS time	2	
time_t (UTC)	4	
time_t (local)	4	

00000000	F4 B8 00 10 8E D0 BC 00-B0 B8 00 00 8E D8 8E C0	ü...-B4-*...@À
00000010	FB BE 00 7C BF 00 06 B9-00 02 F3 A4 EA 21 06 00	ð½·½...·ð½·½...
00000020	00 BE BE 07 38 04 75 0B-83 C6 10 81 FE FE 07 75	·ð½·8-u...E...þþ·u
00000030	F3 EB 16 B4 02 B0 01 BB-00 7C B2 80 82 74 01 88	ð½··*...· ··t..-
00000040	4C 02 CD 13 EA 00 7C 00-00 EB FE 00 00 00 00 00	L·í·è· ··þþ·...
00000050	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000060	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000070	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000080	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000090	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000000a0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000000b0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000000c0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000000d0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000000e0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000000f0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000100	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000110	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000120	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000130	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000140	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000150	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000160	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000170	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000180	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000190	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000001a0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000001b0	00 00 00 00 00 00 00 00-37 E1 27 F5 00 00 00 02-7á·5·
000001c0	03 00 07 7C 3D 7F 80 00-00 00 40 1F 00 00 7C	... *...@...
000001d0	3E 7F 0C F9 2E CB 80 40-1F 00 00 C0 12 00 00 F9	>-ð½·E...À...û
000001e0	2F CB 07 75 6A 4B 80 00-32 00 40 1F 00 00 75	/ð½·u)K...2...û
000001f0	6B 4B 05 FE 7F E8 80 40-51 00 00 C0 26 00 55 AA	kK·þ·ë...@...Û*
00000200	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000210	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

64 byte long
at the bottom

For User Guide, press F1

NUM



File View Mode Help



Evidence Tree

- NONAME [exFAT]
- Partition 2 [600MB]
- NONAME [FAT32]
- Partition 3 [1000MB]
- NTFS_Volume [NTFS]
- Partition 5 [1239MB]
- Unpartitioned Space (basic disk)

Hex Value Interpreter

Type	Size	Value
signed integer	1-8	0
unsigned integer	1-8	0
FILETIME (UTC)	8	-
FILETIME (local)	8	-
DOS date	2	-
DOS time	2	-
time_t (UTC)	4	-
time_t (local)	4	-

File List

Name	Size	Type	Date Modified
------	------	------	---------------

00000000	FA B8 00 10 8E D0 BC 00-B0 B8 00 00 8E D8 8E C0	ü...-B4...-0...À
00000010	FB BE 00 7C BF 00 06 B9-00 02 F3 A4 EA 21 06 00	û...l...-ôhê!..
00000020	00 BE BE 07 38 04 75 0B-83 C6 10 81 FE FE 07 75	û...-u...-E...pp-u
00000030	F3 EB 16 B4 02 B0 01 BB-00 7C B2 80 82 74 01 88	û...-...- ...-t...
00000040	4C 02 CD 13 EA 00 7C 00-00 EB FE 00 00 00 00 00	L...-é...- ...-ép...
00000050	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000060	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000070	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000080	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000090	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000000a0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000000b0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000000c0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000000d0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000000e0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000000f0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000100	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000110	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000120	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000130	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000140	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000150	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000160	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000170	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000180	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000190	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000001a0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000001b0	00 00 00 00 00 00 00 00-37 E1 27 F5 00 00 00 02	...-7â'5...[
000001c0	03 00 07 7C 3D 7F 80 00-00 00 40 1F 00 00 7C	... =...-0...
000001d0	3E 7F 0C F9 2E CB 80 40-1F 00 00 C0 12 00 00 F9	>..û.E@...À...û
000001e0	2F CB 07 75 6A 4B 80 00-32 00 00 40 1F 00 00 75	/È.uJK...2...û...u
000001f0	6B 4B 05 FE 7F E8 80 40-51 00 C0 26 00 55 AA	kK...p...è...Q...À...U*
00000200	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000210	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

century uh is actually indicates us if
we are looking at bootable or

offset 0 (1st byte): non-bootable partition (00)

File View Mode Help



Evidence Tree

- NONAME [exFAT]
- Partition 2 [600MB]
- NONAME [FAT32]
- Partition 3 [1000MB]
- NTFS_Volume [NTFS]
- Partition 5 [1239MB]
- Unpartitioned Space (basic disk)

Hex Value Interpreter

Type	Size	Value
signed integer	1-8	7
unsigned integer	1-8	7
FILETIME (UTC)	8	-
FILETIME (local)	8	-
DOS date	2	-
DOS time	2	-
time_t (UTC)	4	-
time_t (local)	4	-

Byte order: Little endian Big endian

Properties Hex Value Inter... Custom

uh four or uh fifth bytes

uh four or uh fifth bytes

For User Guide, press F1

NUM

offset 4 (5th byte): flag file system type (exfat (07), ext32 (0C), ntfs (07), extended primary partition (05)...)



File View Mode Help



Evidence Tree

...	\.\.\PHYSICALDRIVE2
...	Partition 1 [1000MB]
...	└ NONAME [exFAT]
...	Partition 2 [600MB]
...	└ NONAME [FAT32]
...	Partition 3 [1000MB]
...	└ NTFS Volume [NTFS]

Hex Value Interpreter

Type	Size	Value
signed integer	1-8	8,338,812
unsigned integer	1-8	8,338,812
FILETIME (UTC)	8	-
FILETIME (local)	8	-
DOS date	2	-
DOS time	2	-
time_t (UTC)	4	-
time_t (local)	4	-

Byte order: Little endian Big endian

Properties Hex Value Inter... Custom Content...

Listed: 0 Selected: 0 \.\.\PHYSICALDRIVE2

NUM

the next three bytes are indicating us
chs values of the partition

offset 5-7 (6th - 8th byte): chs value of the partition start

File View Mode Help



Evidence Tree

...	\.\.\PHYSICALDRIVE2
...	Partition 1 [1000MB]
...	└ NONAME [exFAT]
...	Partition 2 [600MB]
...	└ NONAME [FAT32]
...	Partition 3 [1000MB]
...	└ NTFS Volume [NTFS]

Hex Value Interpreter

Type	Size	Value
signed integer	1-8	128
unsigned integer	1-8	128
FILETIME (UTC)	8	-
FILETIME (local)	8	-
DOS date	2	-
DOS time	2	-
time_t (UTC)	4	Thu Jan 1 00:02:00
time_t (local)	4	Thu Jan 1 02:02:00

Byte order: Little endian Big endian

Properties Hex Value Inter... Custom Con... Sel start = 454, len = 4; phy sec = 0

Listed: 0 Selected: 0 \.\.\PHYSICALDRIVE2

NUM

indicates us the
starting sector

offset 8-11 (9th-12th byte): starting sector of partition



File View Mode Help



Evidence Tree

...	\.\.\PHYSICALDRIVE2
...	Partition 1 [1000MB]
...	└ NONAME [exFAT]
...	Partition 2 [600MB]
...	└ NONAME [FAT32]
...	Partition 3 [1000MB]
...	└ NTFS Volume [NTFS]

Hex Value Interpreter

Type	Size	Value
signed integer	1-8	2,048,000
unsigned integer	1-8	2,048,000
FILETIME (UTC)	8	-
FILETIME (local)	8	-
DOS date	2	-
DOS time	2	-
time_t (UTC)	4	Sat Jan 24 16:53:2
time_t (local)	4	Sat Jan 24 18:53:2

Byte order: Little endian Big endian

Properties

Hex Value Inter...

Custom Cont...

Sel start = 458, len = 4; phy rec = 0

Listed: 0 Selected: 0 \.\.\PHYSICALDRIVE2

NUM

and the next four bytes indicate us the
partition length

offset 12-15 (13th-16th byte): partition length in sector (1 sector = 512bytes)



DISK LAYOUT - PARTITION STYLES: MPT

- Partition Type Descriptors represent the partition's file system:
- Examples:
- 0x07 – NTFS
- 0x0C – FAT32
- 0x83 – Linux native
- 0x82 – Linux swap
- 0x05 – Extended Primary Partition
- **More info at: https://en.wikipedia.org/wiki/Partition_type**



uh

DISK LAYOUT - PARTITION STYLES: MBR

- EXTENDED PRIMARY PARTITION – used to overcome 4 partitions MBR limitation
- Type signature in MPT partition record – 0x05 (DoS extended Partition)
Sometimes can also be 0x0F or 0x0C (FAT32 extended)

EXTENDED PRIMARY partition entry – inside Master Boot Record

EXTENDED LOGICAL partition entry – outside Master Boot Record

See all photos

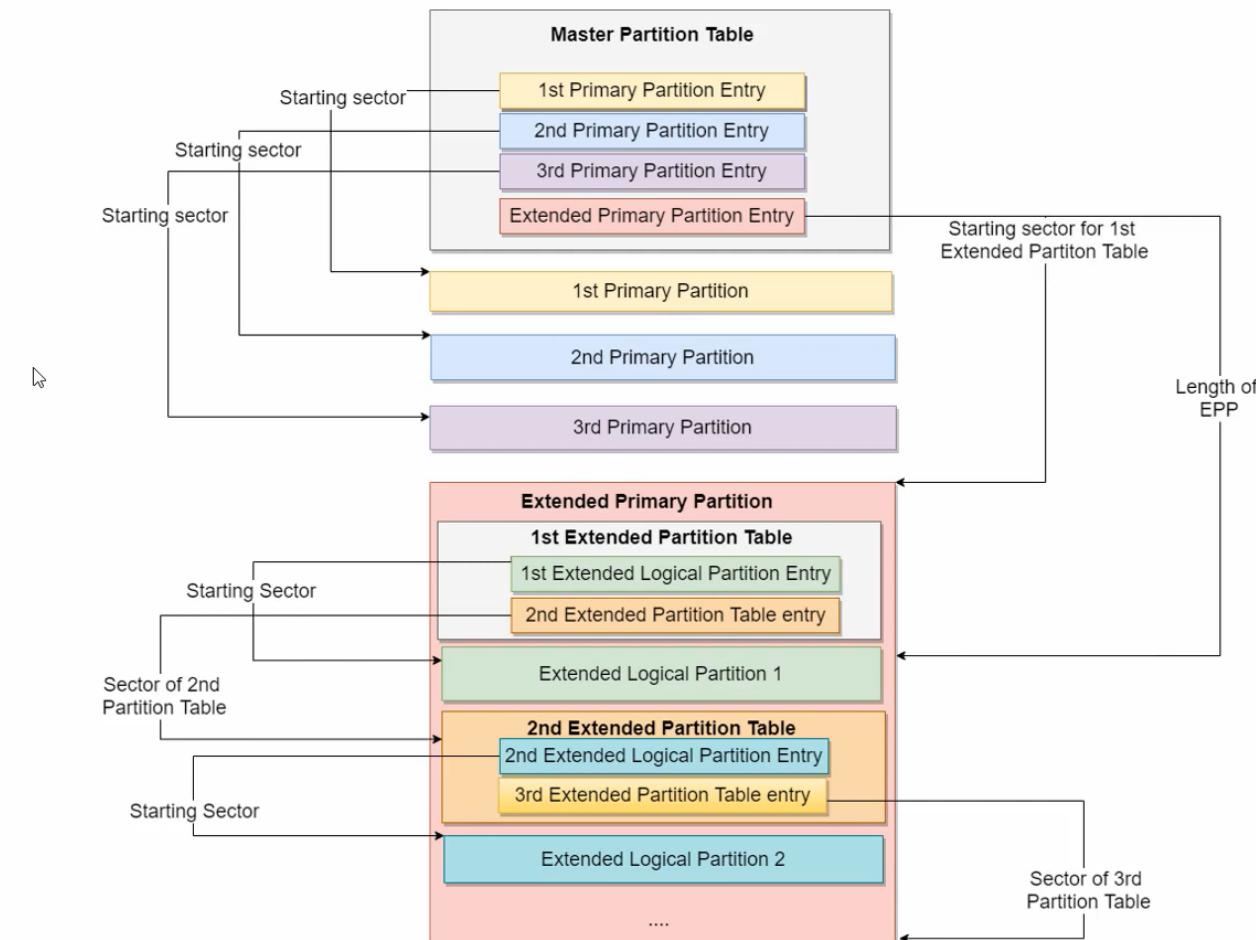
Add to



Edit & Create



Share



okay let's examine this



Type here to search



57°F Cloudy

ENG
ET11:11 AM
9/6/2021

DISK LAYOUT - PARTITION STYLES: GPT

*Also named **GUID partitioning style** – as partitions now have **Global Unique Identifiers assigned to them***

*Compatible with UEFI (**Unified Extensible Firmware Interface**) – a lot of boot data stored on NAND flash or hard drive (boot-time diagnostics, utilities for backup restore or scanning)*

Default partitioning style in Windows (system drive or large external and internal disks)

DISK LAYOUT - PARTITION STYLES: GPT

GPT layout consists of 5 major components, that store all partitioning related metadata:

1. *The “Protective MBR”*
2. *Primary “GUID Partition Table” with GPT header*
3. *GUID partition entries (can accommodate to 128 partitions on the disk)*
4. *Partition area*
5. *Backup area (copies of GUID partition table header and partition entries)*

the gpt partitioning will be created
gpta layout is a bit different

GPT HEADER STRUCTURE:

45	46	49	20	50	41	52	54	00	00	01	00	5C	00	00	00	EFI PART....\...
CB	75	58	F4	00	00	00	00	01	00	00	00	00	00	00	00	�uX�.....
AF	32	CF	1D	00	00	00	00	22	00	00	00	00	00	00	00	�2�....".....
8E	32	CF	1D	00	00	00	00	12	DA	B2	E6	6B	58	90	48	�2�....��kX.H
B0	D0	C4	A5	7A	95	F3	94	02	00	00	00	00	00	00	00	��Yz��".....
80	00	00	00	80	00	00	00	9B	3D	EC	C9	00	00	00	00	�...�...>=i�....

0x00 (8) – signature /x45/x46/x49/x20/x50/x41/x52/x54 – Signature „EFI PART“

0x08 (4) – Revision number for this header

0x0C (4) – GPT header size in bytes (at least 92 bytes, 0x5C)

0x10 (4) – CRC32 checksum of GPT header

0x14 (4) – reserved

0x18 (8) – LBA of the current GPT header structure (sector 1)

0x20 (8) – LBA of backup copy of GPT

0x28 (8) – LBA of start of partition area

0x30 (8) – LBA of the end of partition area (last used partition sector)

0x38 (16) – disk Global Unique Identifier for this disk

0x48 (8) – LBA of the start of the partition tables (usually sector 2)

0x50 (4) – number of possible partition entries (set to 128)

0x54 (4) – size of each entry in the partition table in bytes (set to 128)

0x58 (4) – CRC32 checksum of the partition table

GPT PARTITION ENTRY STRUCTURE:

28 73 2A C1 1F F8 D2 11-BA 4E	00 A0 C9 3E C9 3B	(s*À·øÓ·ºK· È>É;
2A DD 97 3B A8 E8 F2 41-92	57 77 FB 34 6D E5 7F	*Ý ..; "èòA·Wwû4må ·
00 08 00 00 00 00 00 00	-FF 27 08 00 00 00 00 00ý'
01 00 00 00 00 00 80 -45	00 46 00 49 00 20 00·E·F·I· ..
73 00 79 00 73 00 74 00-65	00 6D 00 20 00 70 00	s·y·s·t·e·m· ·p·
61 00 72 00 74 00 69 00-74	00 69 00 6F 00 6E 00	a·r·t·i·t·i·o·n·
00 00 00 00 00 00 00-00	00 00 00 00 00 00 00 00
00 00 00 00 00 00 00-00	00 00 00 00 00 00 00 00

0x00 (16) - partition type GUID

0x10 (16) - Unique partititon GUID

0x20 (8) - starting LBA for the partition

0x28 (8) - ending LBA for the partition

0x30 (8) - partition attribute flags

0x38 (72) - partition name in Unicode

0x30 (8) - partition attribute flags:

0x00 - GPT attribute platform required

0x60 - read only

0x61 - the partition is shadow copy of another partition

0x62 - hidden partition (inaccessible by user)

0x63 - no drive letter assigned



File system

FILE SYSTEM:

- Is a method of organizing (creating changing and retrieving) data on a storage media
- FS provide a method for users to store data in a hierarchy of files and directories.
- Has a defined layout and structure, that consists of:
 - - **FS structural data** – data used by an operating system to search and handle user data (information about file system layout)
 - - **File System level metadata for the files and directories** – stores file or folder parameters and description (e.g., name, timestamps, size, access parameters, etc.)
 - - **File Content** – user/system file content

FILE SYSTEM:

- Structural Data - contains the general file system information that defines the file system layout & parameters (e.g., data unit size(cluster or block), volume layout, media descriptors, allocation bitmaps, etc..)
- Structural Data is usually placed in the beginning of the formatted logical volume (disk)
- Examples:
- FAT, NTFS file systems – 1st sector of a logical volume – **Logical Sector 0**, sometimes called a **Volume Boot Record**
- Linux native ext or xfs filesystems - **Superblock**

FILE SYSTEM:

- File and folder metadata - the data category that describe a file or directory - "data about data" (physical location of the file, LW, LR, LA timestamps, file size, access control information, descriptors, allocation information, a file name)
- Examples:
- FAT file systems - **Directory Entry, File Allocation Table**
- NTFS - **\$MFT (Master File Table)**
- Linux native ext filesystems - **Directory Entry, Inode Table**

FILE SYSTEM:

- Data Related to the performance, resilience and other features of the file more advanced systems
- Examples:
 - journaling and logging of file transactions, data compression or encryption support, disk quotas, pre-allocation of the storage space, etc.)

Disk formatting

DISK FORMATTING:

- Formatting is the process of creating a new file system on a disk partition
- Partitions can be formatted using the quick or full or normal formatting
- During the formatting User specifies :
 1. the file system type of the logical volume (NTFS, FAT32, ext4, xfs, APFS, etc)
 2. the allocation unit size (cluster, or block) - multiple of 512 or sector size

If no cluster size specified by the user, default value is selected based on the size of the partition

assignments
first of all what is the disk formatting

DISK FORMATTING:

During the full formatting:

1. all logical sectors of the partition will be zeroed out -
all previously stored data is gone
2. Structural data and file system metadata (metafiles) are created accordingly to
file system type and allocation unit (cluster) size

During the quick formatting:

1. Structural data and file system metadata (metafiles) are created accordingly to
file system type and allocation unit (cluster) size
2. Old data is not completely gone, but only partially overwritten with the new file
system metadata

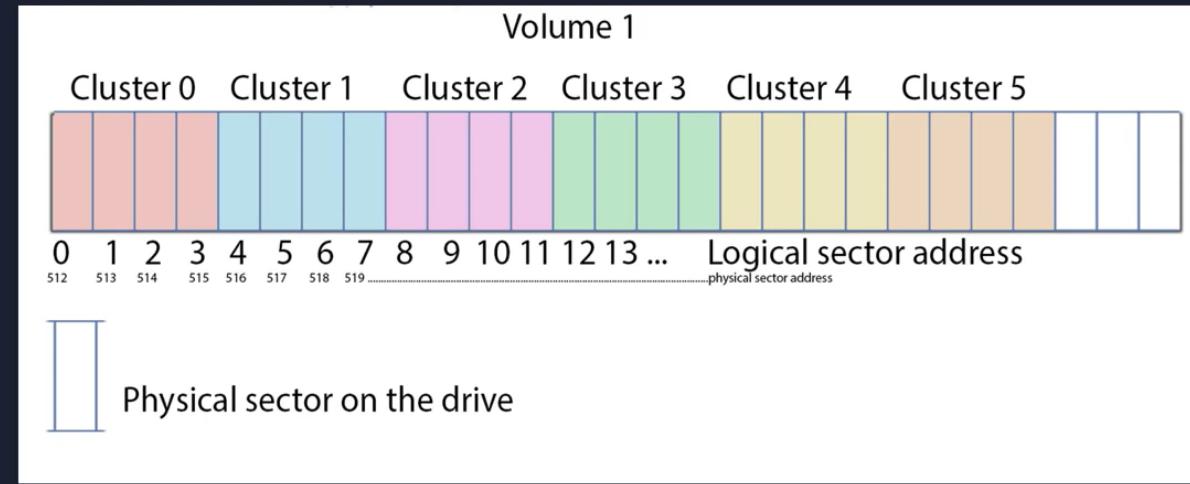
FILE SYSTEM ALLOCATION UNITS:

Called clusters or blocks:

combine several consecutive logical sectors into one addressable unit

cluster or block size can be equal to 1, 2, 4, 8, 16, 32, 64, 128 sectors, or
512b, 1024b, 2048b, 4096b, 8192b, 16384b, 32768b, 65536b

Most commonly 4096 (or 4k) clusters are used in Windows and Linux



FILE SYSTEM ALLOCATION UNITS:

Cluster size represents the smallest amount of disk space that can be used to hold a file.

Example:

FS cluster size is 4096 bytes

User creates a text file with one letter of text (1b) - file logical size

FS allocates 1 cluster to store this file - 4096b - file physical size

4095 bytes are wasted on the disk, and no other file can place its data into the same cluster, until this text file is deleted by the user

FAT file system

FAT FILE SYSTEMS:

- FAT - stands for File Allocation Table
- FAT 12/16 - legacy file system, sometimes used on USB memory sticks up to 1GB
- FAT32 - external storage media devices (USB memory sticks, SD cards,
- exFAT (extended FAT) - larger addressable space than in FAT32, more timestamps and metadata for the files, space pre allocation functionality, possibility to store larger files

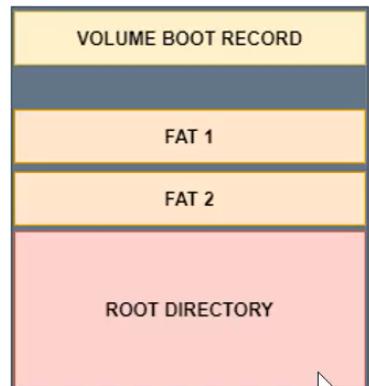
in fat file systems
fat

FAT16 VS FAT32:

FAT 16	FAT32
2 bytes (16 bit) are used for cluster addressing 0xFFFF	3 bytes (28 bit) are used for cluster addressing 0xFF FF FF
Max cluster number is – 65535 clusters, cluster numbering starts from 2	Max cluster number 16,777,214, cluster numbering starts from 2
System Area contains Volume Boot Record, two FAT tables and Root Directory	System Area contains Volume Boot Record, two FAT tables
Data Area contains subdirectories and file data	Data Area contains Root Directory , subdirectories and file data

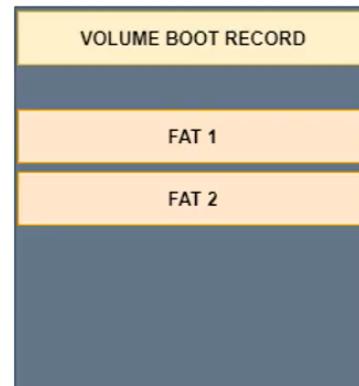
if you compare fat 16 and fat 32 file systems in

FAT 16

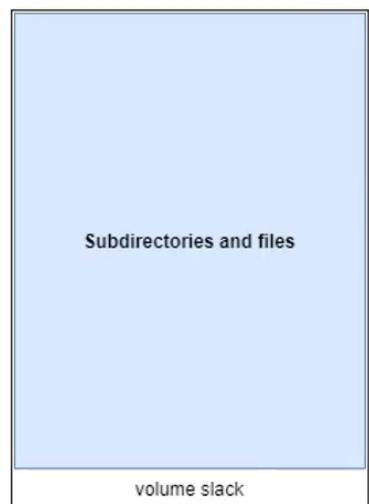


Fixed size
System area

FAT 32



Data area



ROOT DIRECTORY

Subdirectories and files

volume slack

Cluster 2
to
Cluster N



FAT32 VOLUME BOOT RECORD:

00000000	EB	58	90	4D	53	44	4F	53	-35	2E	30	00	02	08	1A	04	ëX	MSDOS5.0ø...?..y.....
00000010	02	00	00	00	00	F8	00	00	-3F	00	FF	00	00	00	00	00	00x..ó.....	
00000020	00	08	78	00	F3	1D	00	00	00	00	00	00	02	00	00	00	00	
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000040	80	00	29	EE	10	80	90	4E	-4F	20	4E	41	4D	45	20	20)i..	-NO NAME	
00000050	20	20	46	41	54	33	32	20	-20	20	33	C9	8E	D1	BC	F4	FAT32	3E-Nº40	
00000060	7B	8E	C1	8E	D9	BD	00	7C	-88	56	40	88	4E	02	8A	56	{ -A-Uñ- -V@-N -V		
00000070	40	B4	41	BB	AA	55	CD	13	-72	10	81	FB	55	AA	75	0A	@'A»-Uí-r-úU-u-		
00000080	F6	C1	01	74	05	FE	46	02	-EB	2D	8A	56	40	B4	08	CD	öÁ-t-pF-é-V@`-í		
00000090	13	73	05	B9	FF	FF	8A	F1	-66	0F	B6	C6	40	66	0F	B6	-s-¹yy-ñf-QE@f-¶		
000000a0	D1	80	E2	3F	F7	E2	86	CD	-C0	ED	06	41	66	0F	B7	C9	Ñ-å?-+â-íÀi-Af-É		
000000b0	66	F7	E1	66	89	46	F8	83	-7E	16	00	75	39	83	7E	2A	f+áf-Fø-~..u9-~*		
000000c0	00	77	33	66	8B	46	1C	66	-83	C0	0C	BB	00	80	B9	01	-w3f-F-f-À-»-..².		
000000d0	00	E8	2C	00	E9	A8	03	A1	-F8	7D	80	C4	7C	8B	F0	AC	·è, ·é"-;ø} Ál-ð-		
000000e0	84	C0	74	17	3C	FF	74	09	-B4	0E	BB	07	00	CD	10	EB	·Àt-<yt-·»-·í-é		
000000f0	EE	A1	FA	7D	EB	E4	A1	7D	-80	EB	DF	98	CD	16	CD	19	i;újëä;} ·ëß í-í-		
00000100	66	60	80	7E	02	00	0F	84	-20	00	66	6A	00	66	50	06	f`-~----·fj-fP-		
00000110	53	66	68	10	00	01	00	B4	-42	8A	56	40	8B	F4	CD	13	Sfh-···B-V@-ðí-		
00000120	66	58	66	58	66	58	66	58-EB	33	66	3B	46	F8	72	03	fXfxfxfxæ3f;Før-			
00000130	F9	EB	2A	66	33	D2	66	0F	-B7	4E	18	66	F7	F1	FE	C2	üë*f3òf-·N-f+fþÅ		
00000140	8A	CA	66	8B	D0	66	C1	EA	-10	F7	76	1A	86	D6	8A	56	·Èf-DfÁé-·v-·ð-V		
00000150	40	8A	E8	C0	E4	06	0A	CC	-B8	01	02	CD	13	66	61	0F	@-èAä-·í-·í-fa-		
00000160	82	74	FF	81	C3	00	02	66	-40	49	75	94	C3	42	4F	4F	-ty-Ã-·f@Iu-ÃBOO		
00000170	54	4D	47	52	20	20	20	20	-00	00	00	00	00	00	00	00	TMGR		
00000180	00	00	00	00	00	00	00	00	-00	00	00	00	00	00	00	00		
00000190	00	00	00	00	00	00	00	00	-00	00	00	00	00	00	00	00		
000001a0	00	00	00	00	00	00	00	00	-00	00	00	00	OD	0A	44	69Di		
000001b0	73	6B	20	65	72	72	6F	72	-FF	0D	0A	50	72	65	73	73	sk errorý-·Press		
000001c0	20	61	6E	79	20	6B	65	79	-20	74	6F	20	72	65	73	74	any key to rest		
000001d0	61	72	74	0D	0A	00	00	00	-00	00	00	00	00	00	00	00	art.....		
000001e0	00	00	00	00	00	00	00	00	-00	00	00	00	00	00	00	00		
000001f0	00	00	00	00	00	00	00	00	-AC	01	B9	01	00	00	55	AA-i-..U^.....		

Bytes 4-8 OEM ID, what OS used for the formatting

Byte 14 – how many sectors in one cluster (512×8) = 4096b cluster size

Bytes 15-16 – offset to the first FAT table

Bytes 29 -32 – sectors between physical sector 0 and this VBR

Offset 0x42 – extended boot signature flag 0x29

Offset 0x43 – Volume label "NO NAME" by default

Offset 52 – Filesystem type

DIRECTORY ENTRIES:

- Store metadata for the files and folders (timestamps, name, physical location, attributes and file size information)
- There are 2 entries exist for each file - base entry (32 bytes, starting with file name converted in DOS 8.3 format and long filename entry - LFN. LFN store file name in UNICODE and occupies as many 32-byte entries as needed to accommodate the name of the file or folder)
- First 32 bytes in ROOT directory store the volume name specified by the user during disk formatting

DIRECTORY ENTRIES:

000	52 45 4D 4F 56 41 42 4C-45 20 20 08 00 00 00 00 00	REMOVABLE
010	00 00 00 00 00 00 67 62-2B 53 00 00 00 00 00 00 00 00gb+S.....
020	42 20 00 49 00 6E 00 66-00 6F 00 0F 00 72 72 00	B .I.n.f.o...rr.
030	6D 00 61 00 74 00 69 00-6F 00 00 00 6E 00 00 00	m.a.t.i.o...n...
040	01 53 00 79 00 73 00 74-00 65 00 0F 00 72 6D 00	.S.y.s.t.e...rm.
050	20 00 56 00 6F 00 6C 00-75 00 00 00 6D 00 65 00	.V.o.l.u...m.e.
060	53 59 53 54 45 4D 7E 31-20 20 20 16 00 4F 66 62	SYSTEM-1 ..-Ofb
070	2B 53 2B 53 00 00 67 62-2B 53 03 00 00 00 00 00 00 00	+S+S..gb+S.....
080	41 53 00 75 00 62 00 66-00 6F 00 0F 00 A1 6C 00	AS.u.b.f.o...;l.
090	64 00 65 00 72 00 5F 00-30 00 00 00 00 00 FF FF	d.e.r._0.....yy
0a0	53 55 42 46 4F 4C 7E 31-20 20 20 10 00 84 04 54	SUBFOL-1T
0b0	2D 53 2D 53 00 00 05 54-2D 53 06 00 00 00 00 00 00 00	-S-S...T-S.....
0c0	41 44 00 6F 00 63 00 75-00 6D 00 0F 00 A7 65 00	AD.o.c.u.m...Se.
0d0	6E 00 74 00 35 00 2E 00-74 00 00 00 78 00 74 00	n.t.5..t...x.t.
0e0	44 4F 43 55 4D 45 7E 31-54 58 54 20 00 29 92 54	DOCUME~1TXT .) .T
0f0	2D 53 2D 53 00 00 93 54-2D 53 10 00 F8 2A 00 00	-S-S...T-S...g^...

so if we look at our disk uh
in cluster

Volume Label (REMOVABLE) -
32b

LFN for "System Volume
Information" folder - 2x32b

Base entry for "System Volume
Information" - 32b

LFN for "Subfolder_1" - 32b

Base entry for Subfolder 1 - 32b

LFN for "Document5.txt" - 32b

Base entry for "Document5.txt" -
32b

DIRECTORY ENTRIES – BASE ENTRY METADATA:

41	44	00	6F	00	63	00	75-00	6D	00	0F	00	A7	65	00	AD	·o·c·u·m···\$e·
6E	00	74	00	35	00	2E	00-74	00	00	00	78	00	74	00	n·t·5·..·t···x·t·	
44	4F	43	55	4D	45	7E	31-54	58	54	20	00	29	92	54	DOCUME~1TXT	·)·T
2D	53	2D	53	00	00	93	54	2D	53	10	00	F8	2A	00	00	-S-S···I-S···ø^··

DOS 8.3 Converted file name (8 UC letters + 3 UC extension)

File attribute (1 byte) – 0x20 or 0010 0000 in binary)

Milliseconds of creation time – 0x29 (41 in decimal)

File Creation TIME - DOS Timestamp (Local time)

File Creation DATE - DOS (Local time)

Last Access DATE - DOS (Local time)

Starting Cluster for the file contents (Higher Bits + Lower Bits)

Content's modification TIME and DATE

FILE SIZE in BYTES - the max value is (0xFF FF FF FF = 4095MB)

File Attributes:

Need to convert to binary:

0000 0001 - read only

0000 0010 - hidden

0000 0100 - system file

0000 1000 - volume label entry

0001 0000 - directory

0010 0000 - archive or normal file

ExFAT File System

EXTENDED FAT FILE SYSTEM

- exFAT :
- 4 bytes used for the cluster addressing
- 8 bytes used for the file size
- Improved performance for free space allocation over free space bitmap
- Timestamp granularity of 10 ms for "Created" and "Modified" timestamps
- Timestamp for Last Access time, FAT32 has only date
- Time Zone value to track difference from UTC time

about uh this file system
so

EX FAT VBR (BOOT SECTOR):

00000000	EB 76 90 45 58 46 41 5	-20 20 20 00 00 00 00 00 00	ev · EXFAT
00000010	00 00 00 00 00 00 00 00	00-00 00 00 00 00 00 00 00 00
00000020	00 00 00 00 00 00 00 00	00-00 00 00 00 00 00 00 00 00
00000030	00 00 00 00 00 00 00 00	00-00 00 00 00 00 00 00 00 00
00000040	20 00 00 00 00 00 00-E0	07 1E 00 00 00 00 00 00 00	à
00000050	80 00 00 00 00 01 00 00-80	01 00 00 19 78 00 00x
00000060	04 00 00 00 46 2F 15 12	00 01 00 00 09 06 01 80	F/.....
00000070	03 00 00 00 00 00 00-33	C9 8E D1 BC F0 7B 8E	3É·Ñø{ ..
00000080	D9 A0 FB 7D B4 7D 8B	F0-AC 98 40 74 0C 48 74 0E	Ù û}·}·ô·@t·Ht·
00000090	B4 0E BB 07 00 CD 10 EB-EF	A0 FD 7D EB E6 CD 16	’·»·í·ëí ý}ëéí·
000000a0	CD 19 00 00 00 00 00-00	00 00 00 00 00 00 00 00 00	í

volume boot record of formatted exfat
this is a

- 0x00 - 2 Jump Code
- 0x03 -10 FS Identifier
- 0x40 - 48 - Volume offset from MBR in sectors
- 0x48 - 4F - total sectors on the volume
- 0x50 - 53 - FAT location
- 0x54 - 57 - FAT size in sectors
- 0x58 - 5B - Bitmap Location
- 0x5C - 5D - cluster count
- 0x60 - 63 - Root directory starting cluster
- 0x64 - 67 volume serial nr
- 0x6D - sectors per cluster as 2^n

DIRECTORY ENTRY:

Offset Hex	Field Definition
x85	Directory Entry Record
x83	Volume Name Record
x82	Up-Case Table Logical Location and Size
x81	Bitmap Logical Location and Size
xC0	Stream Extension
xC1	File Name Extension

- Minimum 3 entries for file or folder:
- 1st starts from 0x85 - Directory entry record, contains timestamps
- 2nd starts from 0xC0 - Stream extension - file size , starting cluster, file name extension parameters
- 3rd starts from 0xC1 - File name extension

a look at the directory entry which is most interesting part for

DIRECTORY ENTRY RECORD:

060	85	03	6E	1A	16	00	00	00-4F	7E	30	53	4F	7E	30	53	..n.....0~0S0~0S
070	4F	7E	30	53	A3	A3	8C	8C-8C	00	00	00	00	00	00	00	0~0\$££.....
080	C0	03	00	19	B8	FF	00	00-00	08	00	00	00	00	00	00	À.....ÿ.....
090	00	00	00	00	7E	00	00	00-00	08	00	00	00	00	00	00
0a0	C1	00	53	00	79	00	73	00-74	00	65	00	6D	00	20	00	Á-S-y-s-t-e-m-
0b0	56	00	6F	00	6C	00	75	00-6D	00	65	00	20	00	49	00	V-o-l-u-m-e- -I-
0c0	C1	00	6E	00	66	00	6F	00-72	00	6D	00	61	00	74	00	Á-n-f-o-r-m-a-t-
0d0	69	00	6F	00	6E	00	00	00-00	00	00	00	00	00	00	00	i-o-n.....

- Off 0x00 - Record Type 0x85
- Off 0x01 - additional 32 records count (3)
- Off 0x04 - DOS flags - (convert to bin. Hid. Sys. Dir.)
- Created tt / dd Modified tt / dd Accessed tt / dd - UTC
- Off 0x14 - 10 MS increments to Created and Modified ts
- Off 0x18 - Time zone offset from UTC
- 0x8C = 1000 1100 : 1000 - TZ flag 1100 - offset in 15 minutes increments: 8+4 = 12, 12*0.25 = 3hr



DIRECTORY ENTRY RECORD, STREAM EXTENSION:

060	85 03 6E 1A 16 00 00 00-4F 7E 30 53 4F 7E 30 53	..n.....0~0\$0~0\$
070	4F 7E 30 53 A3 A3 8C 8C-8C 00 00 00 00 00 00 00 00	0~0\$££-----
080	C0 03 00 19 B8 FF 00 00-00 08 00 00 00 00 00 00 00	À...,\$-----
090	00 00 00 00 7E 00 00 00-00 08 00 00 00 00 00 00 00	-----
0a0	C1 00 53 00 79 00 73 00-74 00 65 00 6D 00 20 00	Á-S-y-s-t-e-m-
0b0	56 00 6F 00 6C 00 75 00-6D 00 65 00 20 00 49 00	V-o-l-u-m-e- -I-
0c0	C1 00 6E 00 66 00 6F 00-72 00 6D 00 61 00 74 00	Á-n-f-o-r-m-a-t-
0d0	69 00 6F 00 6E 00 00 00-00 00 00 00 00 00 00 00	i-o-n-----

- Off 0x00 - Record Type 0xC0 - stream extension
- Off 0x03 - number of Unicode characters in file name (25)
- Off 0x08 - initialized file size in bytes
- Off 0x14 - starting cluster of a file or folder
- Off 0x18 - logical size of the file in bytes
- 0xC1 - File Name extensions

DIRECTORY ENTRY RECORD DELETION:

- 0x85 record type value is changed to 0x05
- 0xC0 record type value is changed to 0x40
- 0xC1 record type value changed to 0x41
- FAT / BITMAP updated with new free clusters information



NTFS

Intro

Native file system for MS Windows

NTFS:

NTFS is capable to use 64bit cluster addressing (default cluster size is 4096 for up to 16TB disk volumes)

Windows limits that number to 32bit cluster addresses

Includes number of advanced features

NTFS ADVANCED FEATURES:

Journaling for better recoverability

File and directory access control

Alternate data streams (multiple data streams)

Disk quotas

Sparse files

File compression

Soft (symbolic) and hard links

Encryption support

Junction points

Metafile backups, "self healing on the run", software-based data redundancy

All metadata stored in metafiles :

Metafile names start with \$ and have no file extension in the name:

\$MFT, \$LogFile, \$UsnJrnl

NTFS:

Metafiles are stored in the root directory of the volume or in \$MetaDirectories

Metafiles can have multiple data streams (named data streams - \$J, \$SDS)

Directory tree is an index B-tree (Balanced Search Tree), stored in \$MFT

NTFS RELATED UPCOMING EPISODES:

Windows File System Forensics - intro to NTFS

Windows File System Forensics - NTFS layout and Volume Boot Record

Windows File System Forensics - Master File Table - \$MFT

Windows File System Forensics - Alternate Data Streams in Windows

Windows File System Forensics - \$LogFile

Windows File System Forensics - analysis of journal \$UsnJrnl

Windows File System Forensics - NTFS Timeline

Boot record

NTFS VOLUME BOOT RECORD:

- Volume starts with a logical sector 0, that stores a Volume Boot Record (part of \$Boot metafile contents)
- Consecutive Logical sectors are combined into clusters
- All space is addressable by means of Logical Cluster Numbers (LCNs)
- \$Boot is in LCN0 and occupies 1 cluster (4096b by default)
- \$Boot has a backup (in very last logical sector of the partition)

VOLUME BOOT RECORD STRUCTURE:

EB 52 90	4E 54 46 53 20-20 20 20	00 02 08 00 00	ëR · NTFS
00 00 00 00 00 00	F8 00 00-3F 00 FF 00 00 00 00 00ø .. ? . ÿ ..	
00 00 00 00 80 00 00 00	FE 07 78 00 00 00 00 00 00ÿ . x ..	
00 00 04 00 00 00 00 00	02 00 00 00 00 00 00 00 00	
F6 00 00 00 01 00 00 00	56 BE DD 0A 03 DE 0A 38	ö VMÝ .. P .. 8	
00 00 00 00 FA 33 C0 8E-D0	BC 00 7C FB 68 C0 07	.. ú3À · Ð¾ · ûhÀ ..	
1F 1E 68 66 00 CB 88 16-0E 00 66 81 3E 03 00 4E	.. hf . È .. f . > .. N		
54 46 53 75 15 B4 41 BB-AA 55 CD 13 72 0C 81 FB	TFSu . 'A» . UI . r .. ú		
55 AA 75 06 F7 C1 01 00-75 03 E9 DD 00 1E 83 EC	U . u .. Á .. u .. éY .. i		
18 68 1A 00 B4 48 8A 16-0E 00 8B F4 16 1F CD 13	.. h .. 'H ô .. í ..		
9F 83 C4 18 9E 58 1F 72-E1 3B 06 0B 00 75 DB A3	.. Á .. X .. rá .. ; .. uÛf ..		
0F 00 C1 2E OF 00 04 1E-5A 33 DB B9 00 20 2B C8	.. Á .. Z3Û .. +È ..		
66 FF 06 11 00 03 16 0F-00 8E C2 FF 06 16 00 E8	fý Áy .. è ..		
4B 00 2B C8 77 EF B8 00-BB CD 1A 66 23 C0 75 2D	K .. +ÈwÍ .. »í .. f #Au ..		
66 81 FB 54 43 50 41 75-24 81 F9 02 01 72 1E 16	f .. úTCPAu\$.. ú .. r ..		
68 07 BB 16 68 52 11 16-68 09 00 66 53 66 53 66	h .. » .. hR .. h .. fSFf ..		
55 16 16 16 68 B8 01 66-61 0E 07 CD 1A 33 C0 BF	U .. h .. fa .. í .. 3Ài ..		
0A 13 B9 F6 0C FC F3 AA-E9 FE 01 90 90 66 60 1E	.. 'ö .. üó .. ép .. f ..		
06 66 A1 11 00 66 03 06-1C 00 1E 66 68 00 00 00	.. f .. f fh ..		
00 66 50 06 53 68 01 00-68 10 00 B4 42 8A 16 0E	.. fP .. Sh .. h .. 'B ..		
00 16 1F 8B F4 CD 13 66-59 5B 5A 66 59 66 59 1F	.. óÍ .. fY[ZfYfY ..		
0F 82 16 00 66 FF 06 11-00 03 16 0F 00 8E C2 FF	.. fý Áy ..		
0E 16 00 75 BC 07 1F 66-61 C3 A1 F6 01 E8 09 00	.. u¾ .. faÁ .. ö .. è ..		
A1 FA 01 E8 03 00 F4 EB-FD 8B F0 AC 3C 00 74 09	;ú .. è .. ôëý .. ð .. < .. t ..		
B4 0E BB 07 00 CD 10 EB-F2 C3 0D 0A 41 20 64 69	' .. » .. í .. èöÁ .. Á di ..		
73 6B 20 72 65 61 64 20-65 72 72 6F 72 20 6F 63	sk read error oc ..		
63 75 72 72 65 64 00 0D-0A 42 4F 4F 54 4D 47 52	curred .. BOOTMGR ..		
20 69 73 20 63 6F 6D 70-72 65 73 73 65 64 00 0D	is compressed ..		
0A 50 72 65 73 73 20 43-74 72 6C 2B 41 6C 74 2B	· Press Ctrl+Alt+ ..		
44 65 6C 20 74 6F 20 72-65 73 74 61 72 74 0D 0A	Del to restart ..		
00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00		
00 00 00 00 00 00 00 00 8A 01-A7 01 BF 01 00 00 55 AA \$.. ? .. U ..		

0x00-02 Jump instructions

0x03-0A - File system name - NTFS

0x0B - 0C - sector size in bytes

0x0D - sectors per cluster

0x28 - 2F - Total number of logical sectors

0x30 - 38 - \$MFT starting logical sector

0x38 - 3F - \$MFT Mirr starting logical sector

0x40 - \$MFT record size, if F6, then 1024b

0x44 - clusters per Index Buffer

0x48 - 4F - Volume Serial Number

0x1FE - Boot signature 0xAA55

MFT

MASTER FILE TABLE - \$MFT

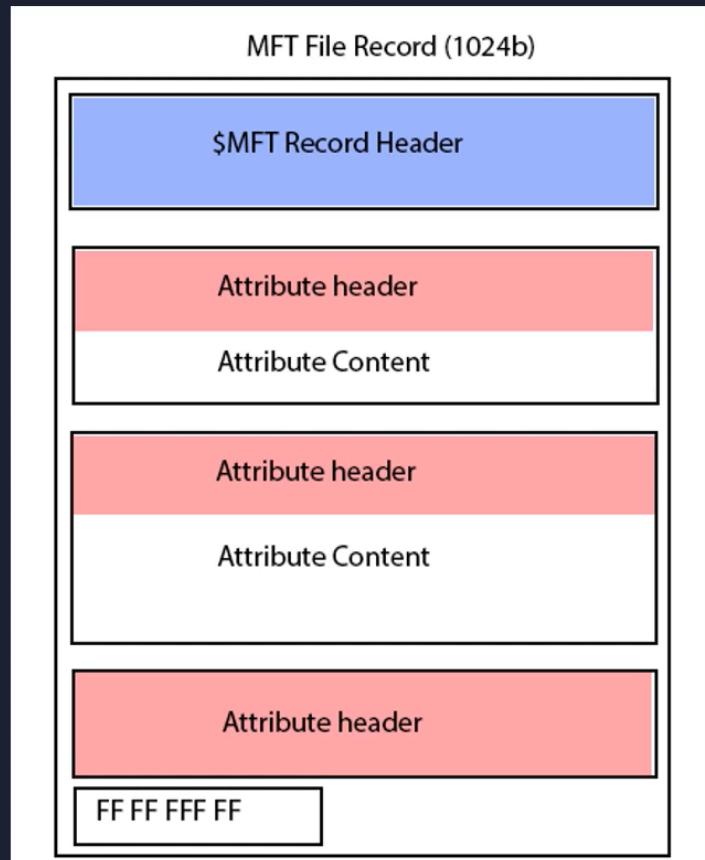
- In NTFS all data stored on a volume is contained in files
- It allows the file system to easily locate and maintain the data and protect all files by a security descriptor
- If sectors goes bad the NTFS can relocate the metafiles
- Master File Table – is the heart of NTFS volume structure
- \$MFT is implemented as an array of file records, starting with signature “FILE0”
- Each File Record is 1024b (4096b can be also defined at volume format time)

MASTER FILE TABLE – FILE RECORDS

0	\$MFT - MFT	23	Unused
1	\$MFTMirr - MFT mirror	24	\$Extend\\$\Quota - Quota information
2	\$LogFile - Log file	25	\$Extend\\$\ObjId - Distributed link tracking information
3	\$Volume - Volume file	26	\$Extend\\$\Reparse - Back references to reparse points
4	\$AttrDef - Attribute definition table	27	\$Extend\\$\RmMetadata - RM metadata directory
5	\ - Root directory	28	\$Extend\\$\RmMetadata\\$Repair - RM repair information
6	\$BitMap - Volume cluster allocation file	29	\$Extend\\$\Deleted - POSIX deleted files
7	\$Boot - Boot sector	30	\$Extend\\$\RmMetadata\\$TxfLog - TxF log directory
8	\$BadClus - Bad-cluster file	31	\$Extend\\$\RmMetadata\\$Txf - TxF metadata directory
9	\$Secure - Security settings file	32	\$Extend\\$\RmMetadata\\$TxfLog\\$Tops - TOPS file
10	\$UpCase - Uppercase character mapping	33	\$Extend\\$\RmMetadata\\$TxfLog\\$TxfLog.blf - TxF BLF
11	\$Extend - Extended metadata directory	34	\$TxfLogContainer00000000000000000000000000000001
		35	\$TxfLogContainer00000000000000000000000000000002

- All records are numbered starting from 0
- 1-st record in \$MFT describes \$MFT itself
- Records 1-12, 23-35 are used for other NTFS metafiles on the volume

MASTER FILE TABLE – FILE RECORD STRUCTURE



- 1024b, starting from "FILE0"
- Stores all the metadata for the file
- In a form File Record header and set of "Attributes"
- If metadata size < 1024b, then only one File Record is used for the file
- If metadata size > 1024b - then several File Records are used – Base Record + Additional Record(s)
- If additional records are used the Attribute content is relocated into additional Record and it's called non-resident attribute

\$MFT – FILE RECORD STRUCTURE:

00000	46 49 4C 45 30 00 03 00-51 15 40 00 00 00 00 00 00	FILE0 ..-Q-@-----
00010	01 00 01 00 38 00 01 00-A0 01 00 00 00 04 00 00	-8-----
00020	00 00 00 00 00 00 00 00-07 00 00 00 00 00 00 00 00	
00030	02 00 00 00 00 00 00 00-10 00 00 00 60 00 00 00 00	
00040	00 00 18 00 00 00 00 00-00-48 00 00 00 18 00 00 00	
00050	6E 12 D9 DB 2A B1 D7 01-6E 12 D9 DB 2A B1 D7 01	n·ÜÜ*±*·n·ÜÜ*±*·
00060	6E 12 D9 DB 2A B1 D7 01-6E 12 D9 DB 2A B1 D7 01	n·ÜÜ*±*·n·ÜÜ*±*·
00070	06 00 00 00 00 00 00 00-00-00 00 00 00 00 00 00 00	
00080	00 00 00 00 00 01 00 00-00 00 00 00 00 00 00 00 00	
00090	00 00 00 00 00 00 00 00-30 00 00 00 68 00 00 00	-0--h-
000a0	00 00 18 00 00 03 00-4A 00 00 00 18 00 01 00	-J-----
000b0	05 00 00 00 00 00 05 00-6E 12 D9 DB 2A B1 D7 01	n·ÜÜ*±*·
000c0	6E 12 D9 DB 2A B1 D7 01-6E 12 D9 DB 2A B1 D7 01	n·ÜÜ*±*·n·ÜÜ*±*·
000d0	6E 12 D9 DB 2A B1 D7 01-00 40 00 00 00 00 00 00 00	n·ÜÜ*±*·@-----
000e0	00 40 00 00 00 00 00 00-00-06 00 00 00 00 00 00 00	@-----
000f0	04 03 24 00 4D 00 46 00-54 00 00 00 00 00 00 00 00	-S·M·F·T-----
00100	80 00 00 00 48 00 00 00-01 00 40 00 00 00 06 00	-H---@-----
00110	00 00 00 00 00 00 00 00-00-3F 00 00 00 00 00 00 00	-?-----
00120	40 00 00 00 00 00 00 00-00-00 00 04 00 00 00 00 00	@-----
00130	00 00 04 00 00 00 00 00-00-00 00 04 00 00 00 00 00	
00140	31 40 00 00 04 00 00 00-B0 00 00 00 50 00 00 00	1@-----P-----
00150	01 00 40 00 00 00 05 00-00-00 00 00 00 00 00 00 00	-@-----
00160	01 00 00 00 00 00 00 00-00-40 00 00 00 00 00 00 00	-@-----
00170	00 20 00 00 00 00 00 00-00-08 10 00 00 00 00 00 00	
00180	08 10 00 00 00 00 00 00-00-31 01 FF FF 03 31 01 26	-1·ÿÿ·1·&-----
00190	00 FC 00 00 00 00 00 00-FF FF FF FF 00 00 00 00 00	-ü-----YYYY-----
001a0	00 00 04 00 00 00 00 00-00-31 40 00 00 04 00 00 00	-1@-----
001b0	B0 00 00 00 50 00 00 00-00-01 00 40 00 00 00 05 00	"-----P-----@-----
001c0	00 00 00 00 00 00 00 00-00-01 00 00 00 00 00 00 00	
001d0	40 00 00 00 00 00 00 00-00-00 20 00 00 00 00 00 00	@-----
001e0	08 10 00 00 00 00 00 00-00-08 10 00 00 00 00 00 00	

\$MFT – FILE RECORD HEADER:

00000	46 49 4C 45	30 00 03 00	00-51 15 40 00 00 00 00 00	FILE0 ... Q @
00010	01 00 01 00	38 00 01 00	-A0 01 00 00 00 04 00 00	... 8
00020	00 00 00 00	00 00 00 00	-07 00 00 00 00 00 00 00
00030	02 00 00 00	00 00 00 00	-10 00 00 00 60 00 00 00
00040	00 00 18 00	00 00 00 00	00-48 00 00 00 18 00 00 00 H

SIGNATURE FILE or BAAD

TRANSACTION ENTRY NUMBER IN \$LogFile

COUNT OF THE FILENAME ATTRIBUTES (HARD LINKS COUNT)

HEADER LENGTH IN BYTES

ALLOCATION STATUS

0x01 - allocated file,
0x00 - deleted file,
0x03 - allocated directory,
0x02 deleted directory

REFERENCE to the BASE RECORD in EXTENDED RECORD

MFT RECORD NUMBER

Timestamps, Standard Information and Filename attributes

\$MFT RECORD ATTRIBUTES:

- Resident attribute consists of:
- Attribute Header (16 bytes)
- Resident value header (6 bytes)
- Value (content) - file metadata

ATTRIBUTES

Attribute ID	Attribute Name	Attribute description & content
10 00 00 00	\$Standard_Information	File permissions, time stamps, security and administrative information, always resident
20 00 00 00	\$Attribute_List	Locations of non-resident attributes
30 00 00 00	\$File_Name	The name of the file – always resident
40 00 00 00	\$Object_ID	GUID (16b) – Globally Unique Identifier
80 00 00 00	\$Data	The actual file's data or pointers to the file's data
90 00 00 00	\$Index_Root	The top-level entry of a sorted tree that lists directory's child files – always resident
A0 00 00 00	\$Index_Allocation	Points to the location of the Index Buffers of a large directory
C0 00 00 00	\$Reparse_Point	Soft link (reparse points), for NTFS junctions and reparse points
00 01 00 00	\$Logged_Utility_Stream	Contains information and keys for encrypted attributes. Used for various purposes, mainly for file encryption management (\$EFS), or transactional NTFS data for the file \$TXF_DATA



STANDARD INFORMATION ATTRIBUTE 0x10

0d515400	46 49 4C 45 30 00 03 00-67 FC 37 3E 10 00 00 00	FILE0...ü7>....
0d515410	1E 00 01 00 38 00 01 00-C0 01 00 00 00 04 00 008...À.....
0d515420	00 00 00 00 00 00 00 00-05 00 00 00 55 54 03 00UT...
0d515430	04 00 00 00 00 00 00 00-10 00 00 00 60 00 00 00
0d515440	00 00 00 00 00 00 00-48 00 00 00 18 00 00 00H....
0d515450	3F 80 17 BA 90 B4 D7 01-62 DE B4 C3 90 B4 D7 01	?...°..‘x·bP'Ã..‘x·
0d515460	4E BB 96 C5 90 B4 D7 01-29 EA A5 5A 12 B9 D7 01	N»·Å..‘x·)ê¥Z..‘x·
0d515470	20 00 00 00 00 00 00 00-00-00 00 00 00 00 00 00
0d515480	00 00 00 00 B5 1C 00 00-00 00 00 00 00 00 00 00p.....
0d515490	60 A7 56 4E 03 00 00 00-30 00 00 00 78 00 00 00	^SVN.....0...x...
0d5154a0	00 00 00 00 00 00 02 00-5A 00 00 00 18 00 01 00Z.....
0d5154b0	4A 34 08 00 00 00 14 00-3F 80 17 BA 90 B4 D7 01	J4.....2...°..‘x·
0d5154c0	3F 80 17 BA 90 B4 D7 01-3F 80 17 BA 90 B4 D7 01	?...°..‘x·-2...°..‘x·
0d5154d0	3F 80 17 BA 90 B4 D7 01-00 00 00 00 00 00 00 00	?...°..‘x.....
0d5154e0	00 00 00 00 00 00 00-20 00 00 00 00 00 00 00
0d5154f0	0C 03 6A 00 6F 00 75 00-72 00 6E 00 61 00 6C 00	..j·o·u·r·n·a·l·
0d515500	33 00 2E 00 74 00 78 00-74 00 00 00 00 00 00 00	3...t·x·t.....

0x0001 - read only

0x0002 - hidden

0x0004 - system file

0x0020, 0x0080 - Archive (normal file)

0x0200 - sparse file

0x0800 - compressed

0x4000 - encrypted (EFS)

0x00 (8) - Date/Time file created on volume

0x08 (8) - Date/Time file modified on volume

0x10 (8) - Date/Time \$MFT content changed

0x18 (8) - Date/Time file last accessed

0x20 (4) - File type flag

0x34 (4) - Security ID (index of permission settings (key to \$SII Index \$SDS data stream in \$Secure))

0x40 (8) - Last Update Sequence Number (\$USN journal), if 0x00 - then journaling is disabled



TIMESTAMPS IN NTFS

- Called FILETIME
- 64-bit unsigned value
- UTC (coordinated universal time)
- Represents the 100 nanosecond intervals since January 1st, 1601
- You need to know the system time zone information to convert it into local time

FILE NAME ATTRIBUTE – 0X30

- There are usually at least 2 FN attributes on system volume:
 1. *POSIX (DOS 8.3)*
 2. *Long File Name - Unicode*

On external volumes POSIX names are disabled by default, can be enabled

FILE NAME ATTRIBUTE – 0x30

079ee400	46 49 4C 45 30 00 03 00-F9 B5 44 2A 11 00 00 00	FILE0...üpD*...
079ee410	22 00 02 00 38 00 01 00-D0 01 00 00 00 04 00 00	"...8...Đ...
079ee420	00 00 00 00 00 00 00-07 00 00 00 B9 E7 01 00ç...
079ee430	03 00 00 00 00 00 00-10 00 00 00 60 00 00 00H...
079ee440	00 00 00 00 00 00 00-48 00 00 00 18 00 00 00	- ô-B*...- ô-B*...
079ee450	20 8E 7C F4 96 DF D7 01-20 8E 7C F4 96 DF D7 01	±..ý-B*...- ô-B*...
079ee460	B1 07 1D FD 96 DF D7 01-20 8E 7C F4 96 DF D7 01
079ee470	20 00 00 00 00 00 00-00 00 00 00 00 00 00 00
079ee480	00 00 00 00 CF OA 00 00-00 00 00 00 00 00 00 00Í...
079ee490	50 EB D4 76 03 00 00-00 30 00 00 00 78 00 00 00	PëÖv...0...x...
079ee4a0	00 00 00 00 00 05 00-5A 00 00 00 18 00 01 00Z...
079ee4b0	4A 34 08 00 00 00 14 00-20 8E 7C F4 96 DF D7 01	J4.....- ô-B*...
079ee4c0	20 8E 7C F4 96 DF D7 01-20 8E 7C F4 96 DF D7 01	- ô-B*...- ô-B*...
079ee4d0	20 8E 7C F4 96 DF D7 01-00 00 00 00 00 00 00 00	- ô-B*.....
079ee4e0	00 00 00 00 00 00 00-20 00 00 00 00 00 00 00 00
079ee4f0	0C 02 54 00 48 00 49 00-53 00 49 00 53 00 7E 00	..T-H-I-S-I-S~..
079ee500	31 00 2E 00 54 00 58 00-54 00 00 00 00 00 00 00	1..T-X-T.....
079ee510	30 00 00 00 78 00 00 00-00 00 00 00 00 00 00 04	0...x...
079ee520	60 00 00 00 18 00 01 00-4A 34 08 00 00 00 14 00J4.....
079ee530	20 8E 7C F4 96 DF D7 01-20 8E 7C F4 96 DF D7 01	- ô-B*...- ô-B*...
079ee540	20 8E 7C F4 96 DF D7 01-20 8E 7C F4 96 DF D7 01	- ô-B*...- ô-B*...
079ee550	00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00
079ee560	20 00 00 00 00 00 00-0F 01 54 00 68 00 69 00T-h-i..
079ee570	73 00 49 00 53 00 4D 00-79 00 44 00 4F 00 43 00	s-I-S-M-y-D-O-C..
079ee580	2E 00 74 00 78 00 74 00-40 00 00 00 28 00 00 00	.t-x-t@...(...
079ee590	00 00 00 00 00 00 06 00-10 00 00 00 18 00 00 00
079ee5a0	3D C4 F3 6F 42 49 EC 11-AD EE A0 A4 C5 11 BC 79	=ÄóoBIi...i mÅ...y
079ee5b0	80 00 00 00 18 00 00 00-00 00 00 18 00 00 00 01
079ee5c0	00 00 00 00 18 00 00 00-FF FF FF FF 82 79 47 11ÿÿÿÿ-yG..
079ee5d0	00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00

Offset in the attribute value field:

0x00 (6) - \$MFT record number of the parent directory

0x08 (8) - File name creation timestamp

0x10 (8) - File name modification timestamp

0x18 (8) - \$MFT Modification Time

0x20 (8) - Last access time



FAT32_over...

Administrator: Windows PowerShell

```
Directory: C:\Users\paffn\Desktop\New folder

Mode           LastWriteTime      Length Name
----           -----          ---- 
-a--- 11/23/2021 10:39 AM        0 ThisIsTheNewestFile.txt

PS C:\Users\paffn\Desktop\New folder> Get-ChildItem -force | Select-Object Mode, Name, CreationTime, LastAccessTime, LastWriteTime | ft

Mode   Name           CreationTime       LastAccessTime       LastWriteTime
----   --           -----          -----          -----
-a--- ThisIsTheNewestFile.txt 11/23/2021 10:39:18 AM 11/23/2021 10:39:18 AM 11/23/2021 10:39:18 AM

PS C:\Users\paffn\Desktop\New folder> $(Get-Item .\ThisIsTheNewestFile.txt).CreationTime=$(Get-Date "01/01/1969 06:00 am")
PS C:\Users\paffn\Desktop\New folder> Get-ChildItem -force | Select-Object Mode, Name, CreationTime, LastAccessTime, LastWriteTime | ft

Mode   Name           CreationTime       LastAccessTime       LastWriteTime
----   --           -----          -----          -----
-a--- ThisIsTheNewestFile.txt 1/1/1969 6:00:00 AM 11/23/2021 10:39:18 AM 11/23/2021 10:39:18 AM

PS C:\Users\paffn\Desktop\New folder> $(Get-Item .\ThisIsTheNewestFile.txt).LastAccessTime=$(Get-Date "01/01/1969 06:00 am")
PS C:\Users\paffn\Desktop\New folder> $(Get-Item .\ThisIsTheNewestFile.txt).LastWriteTime=$(Get-Date "01/01/1969 06:00 am")
PS C:\Users\paffn\Desktop\New folder> Get-ChildItem -force | Select-Object Mode, Name, CreationTime, LastAccessTime, LastWriteTime | ft

Mode   Name           CreationTime       LastAccessTime       LastWriteTime
----   --           -----          -----          -----
-a--- ThisIsTheNewestFile.txt 1/1/1969 6:00:00 AM 1/1/1969 6:00:00 AM 1/1/1969 6:00:00 AM

PS C:\Users\paffn\Desktop\New folder> update
the timestamps
```

CTF

journal3.txt

DigiDoc4 client

ctf2

ctf3

Windows 10 Taskbar: Type here to search, File Explorer, Microsoft Edge, File Explorer, Cloud, Network, Task View, Start button, 33°F, ENG 10:47 AM ET 11/23/2021, Notifications (5)

Understanding where file is located using Data and Attr list attribute

DATA ATTRIBUTE – 0X80

- Header - 0x80 00 00 00
- Used to store information about file size and its content location
- Attribute value can be **resident** - file content is stored directly in the attribute (e.g., small files < 500 b in size)
- or **non-resident** - content location information stored in the "run-lists"

DATA ATTRIBUTE WITH RESIDENT CONTENT

003c2400	46 49 4C 45 30 00 03 00-B8 E7 49 84 08 00 00 00	FILE0...,çI....
003c2410	C2 00 02 00 38 00 01 00-E8 01 00 00 00 04 00 00	Â...8...è.....
003c2420	00 00 00 00 00 00 00-07 00 00 00 09 0F 00 00
003c2430	03 00 00 00 00 00 00-10 00 00 00 60 00 00 00H.....
003c2440	00 00 00 00 00 00 00-48 00 00 00 18 00 00 00H.....
003c2450	BF DD F5 E0 5D 2A D0 01-3D 5D 41 F2 5D 2A D0 01	ÿðà]*Ð-=]Àò]*Ð-
003c2460	3D 5D 41 F2 5D 2A D0 01-BF DD F5 E0 5D 2A D0 01	=]Àò]*Ð-ÿðà]*Ð-
003c2470	20 00 00 00 00 00 00-00 00 00 00 00 00 00 00
003c2480	00 00 00 00 D1 11 00 00-00 00 00 00 00 00 00 00	...Ñ.....
003c2490	B8 4A 6C CA 01 00 00 00-30 00 00 00 78 00 00 00	,J1È...0...x...
003c24a0	00 00 00 00 00 05 00-5A 00 00 00 18 00 01 00Z.....
003c24b0	00 02 04 00 00 00 0A 00-BF DD F5 E0 5D 2A D0 01ÿðà]*Ð-
003c24c0	BF DD F5 E0 5D 2A D0 01-BF DD F5 E0 5D 2A D0 01	ÿðà]*Ð-ÿðà]*Ð-
003c24d0	BF DD F5 E0 5D 2A D0 01-00 00 00 00 00 00 00 00	ÿðà]*Ð.....
003c24e0	00 00 00 00 00 00 00-20 00 00 00 00 00 00 00 00
003c24f0	0C 02 41 00 42 00 52 00-41 00 4B 00 41 00 7E 00	..A·B·R·A·K·A~..
003c2500	31 00 2E 00 54 00 58 00-54 00 74 00 78 00 74 00	1..·T·X·T·t·x·t·
003c2510	30 00 00 00 78 00 00 00-00 00 00 00 00 00 04 00	0...x.....
003c2520	60 00 00 00 18 00 01 00-00 02 04 00 00 00 0A 00	'.....
003c2530	BF DD F5 E0 5D 2A D0 01-BF DD F5 E0 5D 2A D0 01	ÿðà]*Ð-ÿðà]*Ð-
003c2540	BF DD F5 E0 5D 2A D0 01-BF DD F5 E0 5D 2A D0 01	ÿðà]*Ð-ÿðà]*Ð-
003c2550	00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00
003c2560	20 00 00 00 00 00 00-00 0F 01 61 00 62 00 72 00a·b·r·
003c2570	61 00 6B 00 61 00 64 00-61 00 62 00 72 00 61 00	a·k·a·d·a·b·r·a·
003c2580	2E 00 74 00 78 00 74 00-40 00 00 00 28 00 00 00	.·t·x·t·@··(··
003c2590	00 00 00 00 00 00 06 00-10 00 00 00 18 00 00 00	·
003c25a0	46 52 2C 08 36 96 E4 11-A8 18 DB 6C 61 6F F3 0F	FR,-6·ä..Ûlaoó..
003c25b0	80 00 00 00 30 00 00 00-00 00 18 00 00 00 01 00	...0.....
003c25c0	16 00 00 00 18 00 00 00-00 61 62 72 61 6B 61 64 61abarakada
003c25d0	62 72 61 20 73 69 6D 73-61 6C 61 62 69 6D 00 00	bra simsalabim..
003c25e0	FF FF FF FF 82 79 47 11-00 00 00 00 00 00 00 00 03 00	ÿÿÿ·yG.....
003c2600	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

Offset in the attribute value field:

0x00 (4) - Data attribute ID - 0x80

0x05 (2) - size of attribute in bytes

0x08 (1) - content flag:

0x00 - resident

0x01 - nonresident

DATA ATTRIBUTE WITH NONRESIDENT CONTENT

09c00	46 49 4C 45 30 00 03 00-36 50 20 00 00 00 00 00 FILE0 ..-6P
09c10	01 00 02 00 38 00 01 00-E0 01 00 00 00 04 00 008 ..-â.....
09c20	00 00 00 00 00 00 00 00-04 00 00 00 27 00 00 00
09c30	03 00 00 00 00 00 00 00-10 00 00 00 60 00 00 00
09c40	00 00 00 00 00 00 00 00-04 00 00 00 18 00 00 00
09c50	D3 02 FE 9B 88 10 D0 01-B8 AE FA 7B 44 04 CA 01 Ó·p...-D...æ[D·È..
09c60	05 78 FE 9B 88 10 D0 01-D3 02 FE 9B 88 10 D0 01 -xp...-D·O·p...-D·.
09c70	20 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
09c80	00 00 00 05 01 00 00-00 00 00 00 00 00 00 00
09c90	00 00 00 00 00 00 00 00-03 00 00 00 78 00 00 00
09ca0	00 00 00 00 00 00 03 00-05A 00 00 00 18 00 01 00
09cb0	26 00 00 00 00 01 00-00 00 00 00 00 00 10 D0 01
09cc0	D3 02 FE 9B 88 10 D0 01-D3 02 FE 9B 88 10 D0 01 Ó·p...-D·Ó·p...-D·.
09cd0	D3 02 FE 9B 88 10 D0 01-00 6C 0D 00 00 00 00 00
09ce0	00 00 00 00 00 00 00 00-00 20 00 00 00 00 00 00
09cf0	OC 02 43 00 48 00 52 00-59 00 53 00 41 00 7E 00 ..-C·H·R·Y·S·A~..
09d00	31 00 2E 00 4A 00 50 00-47 00 6D 00 2E 00 6A 00 1...J·P·G·m...j..
09d10	30 00 00 00 80 00 00 00-00 00 00 00 00 00 00 02 00 0.....
09d20	64 00 00 00 18 00 00 01-00 26 00 00 00 00 00 01 00 d.....g..
09d30	D3 02 FE 9B 88 10 D0 01-D3 02 FE 9B 88 10 D0 01
09d40	D3 02 FE 9B 88 10 D0 01-D3 02 FE 9B 88 10 D0 01
09d50	00 6C 0D 00 00 00 00 00-00 00 00 00 00 00 00 00 ..-1.....
09d60	20 00 00 00 00 00 00 00-00 11 01 43 00 68 00 72 00
09d70	79 00 73 00 61 00 6E 00-74 00 68 00 65 00 6D 00 y·s·a·n·t·h·e·m..
09d80	75 00 6D 00 2E 00 6A 00-70 00 67 00 00 00 00 00 ..u·m..j·p·g..
09d90	80 00 00 00 1A 00 00 00-01 00 00 00 00 00 00 01 00 ..-H.....
09da0	00 00 00 00 00 00 00 00-00 5A 03 00 00 00 00 00 ..-Z.....
09db0	40 00 00 00 00 00 00 00-00 00 6C 0D 00 00 00 00 ..@.....1.....
09dc0	22 6B 0D 00 00 00 00 00-22 6B 0D 00 00 00 00 00 .."k....."k..
09dd0	32 5B 03 CB EB 04 00 FF-FF FF FF FF 82 79 47 11 2[.È·È..VVVVV·yG..
09de0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
09df0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 03

- Content non-resident flag (0x01)
- Attribute length in bytes (0x48) - 72b
- Starting VCN number (0)
- Ending VCN on the run list (858)
- offset to the run list in bytes (64)
- physical size of the file in bytes(allocated clusters in b)
- logical size of the file in bytes
- initialized size for the content in bytes
- RUN LIST

- Flag 0x01 In the attribute header offset 0x08 points to non-resident content
- Content structure:
 - 0x10 (8) - Starting VCN of the run list
(Virtual Cluster Number relative to start of data)
 - 0x18 (8) - Ending VCN of the run list
 - 0x20 (2) - Offset to the run list from the start of the attribute
- 0x28 (8) - Physical size of the file
- 0x30 (8) - Logical size of the file
- 0x38 (8) - Initialized size of the file
- ~ Run list



\$DATA ATTRIBUTE RUN LISTS

08c00	46 49 4C 45 30 00 03 00-0B 3F 20 00 00 00 00 00 00	FILE0-...?
08c10	01 00 02 00 38 00 01 00-38 02 00 00 00 04 00 00	...-8-8.....
08c20	00 00 00 00 00 00 00 00-06 00 00 00 23 00 00 00#....
08c30	03 00 00 00 00 00 00 00-00-10 00 00 00 60 00 00 00'....
08c40	00 00 00 00 00 00 00 00-00-48 00 00 18 00 00 00H....
08c50	A6 A6 15 59 88 10 D0 01-E8 E9 D2 7D AF 7D CC 01	.Y.-D.-ééò)ì.
08c60	43 D5 4C 60 88 10 D0 01-A6 A6 15 59 88 10 D0 01	CÖL`-D- .Y.-D.
08c70	20 00 00 00 00 00 00 00-00-00 00 00 00 00 00 00 00
08c80	00 00 00 00 05 01 00 00-00 00 00 00 00 00 00 00 00
08c90	00 00 00 00 00 00 00 00-00-30 00 00 00 78 00 00 000-x....
08ca0	00 00 00 00 00 05 00 00-00-5A 00 00 00 18 00 01 00Z....
08cb0	24 00 00 00 00 00 01 00-A6 A6 15 59 88 10 D0 01	\$-..... .Y.-D.
08cc0	E8 E9 D2 7D AF 7D CC 01-B7 CD 15 59 88 10 D0 01	ééò)ì.-í.Y.-D.
08cd0	A6 A6 15 59 88 10 D0 01-00 80 01 00 00 00 00 00 00	.Y.-D....
08ce0	C0 7C 01 00 00 00 00 00-00-20 00 00 00 00 00 00 00	À
08cf0	0C 02 55 00 53 00 45 00-46 00 55 00 4C 00 7E 00	-U-S-E-F-U-L~.
08d00	31 00 2E 00 50 00 44 00-46 00 74 00 65 00 72 00	1...P-D-F-t-e-r.
08d10	30 00 00 00 D8 00 00 00-00 00 00 00 00 00 04 00	0...Ø.....
08d20	C0 00 00 00 18 00 01 00-00-24 00 00 00 00 00 01 00	À.....\$....
08d30	A6 A6 15 59 88 10 D0 01-E8 E9 D2 7D AF 7D CC 01	.Y.-D.-ééò)ì.
08d40	B7 CD 15 59 88 10 D0 01-A6 A6 15 59 88 10 D0 01	-í.Y.-D- .Y.-D.
08d50	00 80 01 00 00 00 00 00-C0 7C 01 00 00 00 00 00À
08d60	20 00 00 00 00 00 00 00-00-3F 01 55 00 73 00 65 00?U-s-e.
08d70	66 00 75 00 6C 00 2E 00-43 00 6F 00 6D 00 70 00	f-u-l-.C-o-m-p.
08d80	75 00 74 00 65 00 72 00-2E 00 46 00 6F 00 72 00	u-t-e-r..F-o-r.
08d90	65 00 6E 00 73 00 69 00-63 00 73 00 2E 00 54 00	e-n-s-i-c-s.-T.
08da0	6F 00 6F 00 6C 00 73 00-28 00 65 00 62 00 6F 00	o-o-l-s-(e-b-o-
08db0	6F 00 6B 00 2E 00 6D 00-6F 00 6E 00 73 00 74 00	o-k..-m-o-n-s-t.
08dc0	65 00 72 00 2E 00 62 00-6C 00 6F 00 67 00 73 00	e-r..-b-l-o-g-s.
08dd0	70 00 6F 00 74 00 2E 00-63 00 6F 00 6D 00 29 00	p-o-t..-c-o-m-).
08de0	2E 00 70 00 64 00 66 00-80 00 00 00 48 00 00 00	.p-d-f...-H...
08df0	01 00 00 00 00 00 01 00-00-00 00 00 00 00 00 03 00@....
08e00	SF 00 00 00 00 00 00 00-00-40 00 00 00 00 00 00 00À
08e10	00 80 01 00 00 00 00 00-00-C0 7C 01 00 00 00 00 00
08e20	CO 7C 01 00 00 00 00 00-31 60 D7 EA 04 00 FF FF	À1*xè..yy
08e30	FF FF FF 82 79 47 11-00 00 00 00 00 00 00 00 00	yyyy.yG.....
08e40	00 00 00 00 00 00 00 00-00-00 00 00 00 00 00 00 00
08e50	00 00 00 00 00 00 00 00-00-00 00 00 00 00 00 00 00
08e60	00 00 00 00 00 00 00 00-00-00 00 00 00 00 00 00 00
08e70	00 00 00 00 00 00 00 00-00-00 00 00 00 00 00 00 00
08e80	00 00 00 00 00 00 00 00-00-00 00 00 00 00 00 00 00

Run list consists of:

Header - 1 byte

Header's right nibble (1) - number of bytes used to store the fragment length in clusters

Header's left nibble (3) - number of bytes used to store starting cluster number for the file content

In our example:

There is only one run list - file content is not fragmented

File data length in clusters - 0x60 or 96 clusters

File starts at 0x4EAD7 or 322263 cluster

ObjectID attribute

OBJECT ID ATTRIBUTE -0X40

- Header 0x40 00 00 00
- Contains GUID-1 value
- Resident, but optional (not always there)
- NTFS link-tracking support is based on object ID attribute (files can be opened by their object IDs rather than their file name)
- An application can assign an object ID to a file with file system control code FSCTL_SET_OR_GET_OBJECT_ID, or delete it with FSCTL_DELETE_OBJECT_ID
- File object ID-s are also stored in \$ObjId metafile (object identifier file)

OBJECT ID ATTRIBUTE -0X40

- ObjectID attribute stores 16 byte long GUID value for a file or folder
- GUID-1 value of file's object ID can be converted into MAC address and GUID timestamp (OS boot time and date prior the file creation)
- NTFS keeps the correspondence between Object IDs and \$MFT record number of the file in \$O index of \$ObjId metafile

Directory Tree Indexes

DIRECTORY TREE STRUCTURE IN NTFS

- NTFS uses indexes (attributes that are stored in a sorted order) to compose the directory tree and track changes in the directories
- The file system elements that store the NTFS directory indexes are \$Index_Root, \$Index_Allocation attributes and \$Index_Buffers
- Directory tree is organized using a B-Tree concept based on the filenames
- Indexes store the directory names along with directory related timestamps



Evidence Tree

...	C:\
...	NONAME [NTFS]
...	[orphan]
...	[root]
...	\$BadClus
...	\$Bitmap
...	\$Extend
...	\$Recycle.Bin
...	\$Secure
...	\$UpCase
...	\$WinREAgent
...	Documents and Settings
...	PerfLogs
...	Program Files
...	Program Files (x86)
...	ProgramData
...	Recovery
...	System Volume Information
...	Users
...	All Users
...	Default
...	Default User
...	forensic

Hex Value Interpreter

Type	Size	Value
signed integer	1-8	
unsigned integer	1-8	
FILETIME (UTC)	8	
FILETIME (local)	8	

Byte order: Little endian Big endian

Properties Hex Value Int... Custom Cont...

For User Guide, press F1



Type here to search

Index_Root: \$I30 (0x900000)

Timestamp of File: offset

so if you see the stream name inside the
mft then you are looking



^ ⌂ ⌂ ENG
ET 8:19 AM
2/4/2022

Name	Size	Type	Date Modified
\$Boot	8	Regular File	1/31/2022 8:11:09 PM
\$I30	4	NTFS Index All...	2/2/2022 11:24:00 AM
\$LogFile	65,536	Regular File	1/31/2022 8:11:09 PM
SMFT	271,104	Regular File	1/31/2022 8:11:09 PM
SMFTMirr	4	Regular File	1/31/2022 8:11:09 PM
SSecure	1	Regular File	1/31/2022 8:11:09 PM
DATA DATA	1	NTFS	2/2/2022 11:24:00 AM
0634b4d0	49 6D 47 7E 9D 19 D8 01-00 00 00 00 00 00 00 00	ImG~··0.....	
0634b4e0	00 00 00 00 00 00 00-00 00 00 10 00 00 00 00 00	
0634b4f0	06 03 71 00 77 00 65 00-72 00 74 00 79 00 00 00	..q-w-e-r-t-y...	
0634b500	40 00 00 00 28 00 00 00-00 00 00 00 00 00 00 05 00	@...(...	
0634b510	10 00 00 00 18 00 00 00-00 1A 6F 6F 54 94 82 EC 11ooT-i...	
0634b520	B7 DC A0 A4 C5 11 BC 7C-90 00 00 00 20 01 00 00	Ü HÀ~!.....	
0634b530	00 04 18 00 00 00 01 00-00 01 00 00 20 00 00 00	
0634b540	24 00 49 00 33 00 30 00-30 00 00 00 01 00 00 00	\$·I·3·0·0·	
0634b550	00 10 00 00 01 00 00 00-10 00 00 00 F0 00 00 00·8...	
0634b560	F0 00 00 00 00 00 00 00-00 12 ED 00 00 00 00 0A 00	8.....i...	
0634b570	68 00 54 00 00 00 00 00-00 2D 8D 01 00 00 00 06 00	h·T.....	
0634b580	A6 32 BE 84 9D 19 D8 01-A6 32 BE 84 9D 19 D8 01	;2%···0·;2%···0·	
0634b590	D2 6F CB 88 9D 19 D8 01-A6 32 BE 84 9D 19 D8 01	ÓoÈ···0·;2%···0·	
0634b5a0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
0634b5b0	20 00 00 00 00 00 00 00-09 03 61 00 73 00 64 00a·s·d·	
0634b5c0	66 00 67 00 2E 00 74 00-78 00 74 00 54 00 58 00	f·g..·t·x·t·T·X·	
0634b5d0	94 80 01 00 00 00 0A 00-68 00 54 00 00 00 00 00h·T.....	
0634b5e0	2D 8D 01 00 00 00 06 00-A6 32 BE 84 9D 19 D8 01	-.....;2%···0·	
0634b5f0	2F 60 A7 8B 9D 19 D8 01-BE E9 59 8F 9D 19 04 00	/`\$···0·%éY.....	
0634b600	2F 60 A7 8B 9D 19 D8 01-00 00 00 00 00 00 00 00	/`\$···0.....	
0634b610	00 00 00 00 00 00 00 00-00 20 00 00 00 00 00 00	
0634b620	09 03 7A 00 78 00 63 00-76 00 62 00 2E 00 74 00	..z·x·c·v·b..·t·	
0634b630	78 00 74 00 54 00 58 00-00 00 00 00 00 00 00 00	x·t·T·X.....	
0634b640	10 00 00 00 02 00 00 00-00 FF FF FF FF 82 79 47 11YYYY·yG·	
0634b650	FF FF FF 82 79 47 11-70 00 5A 00 00 00 00 00	YYYY·yG·p·Z.....	

AccessData FTK Imager 4.5.0.3
020 Windows NTFS MFT Part 5 Understanding Directory Tree Indexes

Evidence Tree | File List | Hex | Text | DIR | ?

Evidence Tree

- [root]
 - \$BadClus
 - \$Bitmap
 - \$Extend
 - \$Recycle.Bin
 - \$Secure
 - \$UpCase
 - \$WinREAgent
 - Documents and Settings
 - PerfLogs
 - Program Files
 - Program Files (x86)
 - ProgramData

File List

Name	Size	Type	Date Modified
SMFT	271,104	Regular File	1/31/2022 8:11:09 PM
0634b500	40 00 00 00 28 00 00 00-00 00 00 00 00 00 05 00	@	
0634b510	10 00 00 00 18 00 00 00-1A 6F 6F 54 94 82 EC 11ooT-i.	
0634b520	B7 DC A0 A4 C5 11 BC 7C-90 00 00 00 58 00 00 00	·Ü ·A ·4 ·X ..	
0634b530	00 04 18 00 00 00 08 00-38 00 00 00 20 00 00 008	
0634b540	24 00 49 00 33 00 30 00-30 00 00 00 01 00 00 00	\$-I-3-0-0	
0634b550	00 10 00 00 01 00 00 00-10 00 00 00 28 00 00 00(.....	
0634b560	28 00 00 00 01 00 00 00-00 00 00 00 00 00 00 00	(.....	
0634b570	18 00 00 00 03 00 00 00-00 00 00 00 00 00 00 00	
0634b580	A0 00 00 00 50 00 00 00-01 04 40 00 00 00 06 00P	
0634b590	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
0634b5a0	48 00 00 00 00 00 00 00-00 00 10 00 00 00 00 00	H	
0634b5b0	00 10 00 00 00 00 00 00-00 00 10 00 00 00 00 00	
0634b5c0	24 00 49 00 33 00 30 00-31 01 D8 9B 23 00 00 00	\$-I-3-0-1-0-# ..	
0634b5d0	B0 00 00 00 28 00 00 00-00 00 04 18 00 00 00 07 00	^	
0634b5e0	08 00 00 00 20 00 00 00-24 00 49 00 33 00 30 00\$-I-3-0-.	
0634b5f0	01 00 00 00 00 00 00 00-00 FF FF FF 82 79 0B 00YYYY:Y..	
0634b600	C2 6B 92 33 A0 19 D8 01-00 00 00 00 00 00 00 00	Äk-3 -0	
0634b610	00 00 00 00 00 00 00 00-00 20 00 00 00 00 00 00	
0634b620	52 01 64 00 61 00 66 00-6B 00 6A 00 73 00 64 00	R-d-a-f-k-j-s-d-	
0634b630	6B 00 66 00 6C 00 6A 00-61 00 73 00 64 00 66 00	k-f-l-j-a-s-d-f-	
0634b640	6C 00 6B 00 F6 00 61 00-73 00 F6 00 64 00 67 00	l-k-ö-a-s-ö-d-g-	
0634b650	6C 00 6B 00 6A 00 61 00-73 00 67 00 6B 00 6C 00	l-k-j-a-s-g-k-l-	
0634b660	EA 00 70 00 11 00 6B 00-6C 00 67 00 FA 00 61 00	j-s-a-k-l-g-j-a-	
0634b670	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	s-d-k-l-f-j-a-d-	
0634b680	6B 00 6C 00 66 00 6A 00-61 00 64 00 73 00 6B 00	k-l-f-j-a-d-s-k-	

that will store all uh you know all the file names uh inside this directory

Hex Value Interpreter

Type	Size	Value
signed integer	1-8	-96
unsigned integer	1-8	160
FILETIME (UTC)	8	.
FILETIME (local)	8	.
DOS date	2	.
DOS time	2	.
time_t (UTC)	4	.

Byte order: Little endian Big endian

Properties Hex Value Inter... Custom

For User Guide, press F1

Windows Taskbar: Type: 10:40/a/14 8:34 AM ET 4/2022

Index_Allocation: (0xA00000)

offset 32 (0x48): size of attribute

after that is runlist

- Index_Allocation point to Index_Buffer

[orlikoski/CyLR: CyLR - Live Response Collection Tool \(github.com\)](https://orlikoski/CyLR: CyLR - Live Response Collection Tool (github.com))

[blueangel's ForensicNote \(google.com\)](#)

[Run FTK Imager from a flash drive \(Imager Lite\) : Support Portal \(freshdesk.com\)](#)

[Extracted_metafiles - Google Drive](#)

..

EXT

ext2 ext3 and ext4 filesystems

EXT(ENDED) FILE SYSTEMS

EXT2 - 1993 , first stable EXT FS

- up to 2 TB files and 32 TB file systems
- uses block groups and allocation bitmaps to limit file system fragmentation
- E2fsprogs - ext2 utilities
- fsck - FS check and repair utility
- libext2fs - general library to access ext2

let's
first take a look at some facts related

EXT(ENDED) FILE SYSTEMS

ext3 FS - 2001 (new version of ext2)

Added features:

- Journalling
- Hashed btrees
- ACL's and Extended Attributes
- backward and forward compatibility with ext2

it's not suitable for the modern
operating systems

EXT(ENDED) FILE SYSTEMS

ext4 FS

Added features:

- Use of extents instead of indirect blocks
- Delayed Allocation (lazy write)
- fallocate() - allocation blocks in advance (free blocks indication without wiping)
- subsecond timestamps
- option to enable or disable FS journal
- tons of configurable options

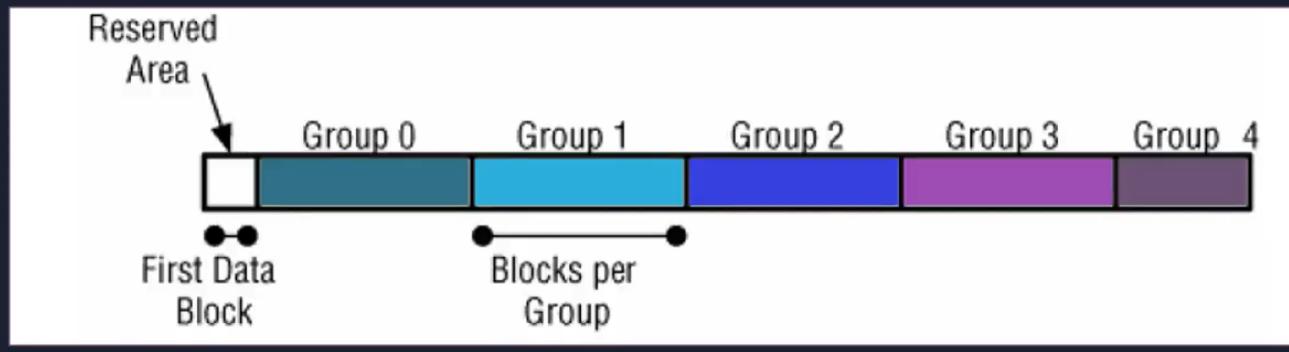
actually for tracking the file
data that's why the x4 file system was

EXT FILE SYSTEM ELEMENTS

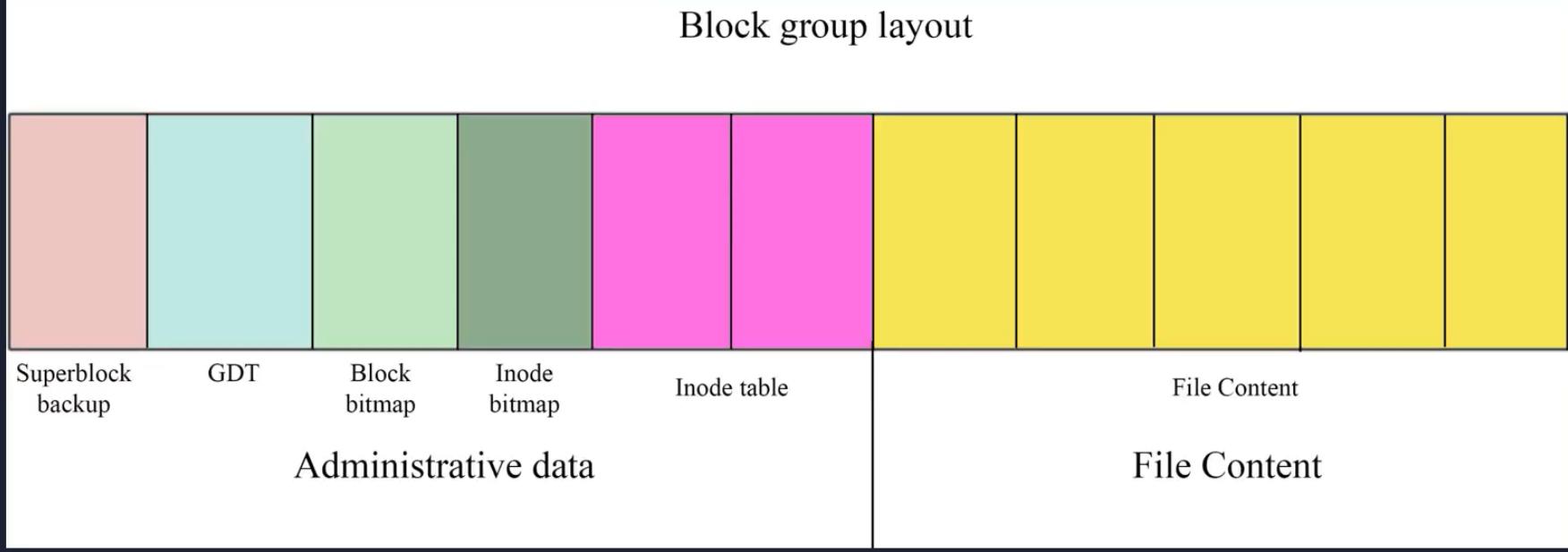
- Main elements:
- Blocks = clusters
- Block groups
- Superblock
- Group Descriptor Table
- Inodes
- Directory Entries
- Block Bitmap

EXT FILE SYSTEM ELEMENTS

- The layout of the file system starts with the **optional reserved area**, and the remainder of the allocated space devided into sections, which are called *block groups*.
- *Block groups* (except for the last) contain the same number of blocks, which are used to store file names, metadata and file content
- The extX file system layout is stored in **superblock** in the beginning of the file system



Block group layout



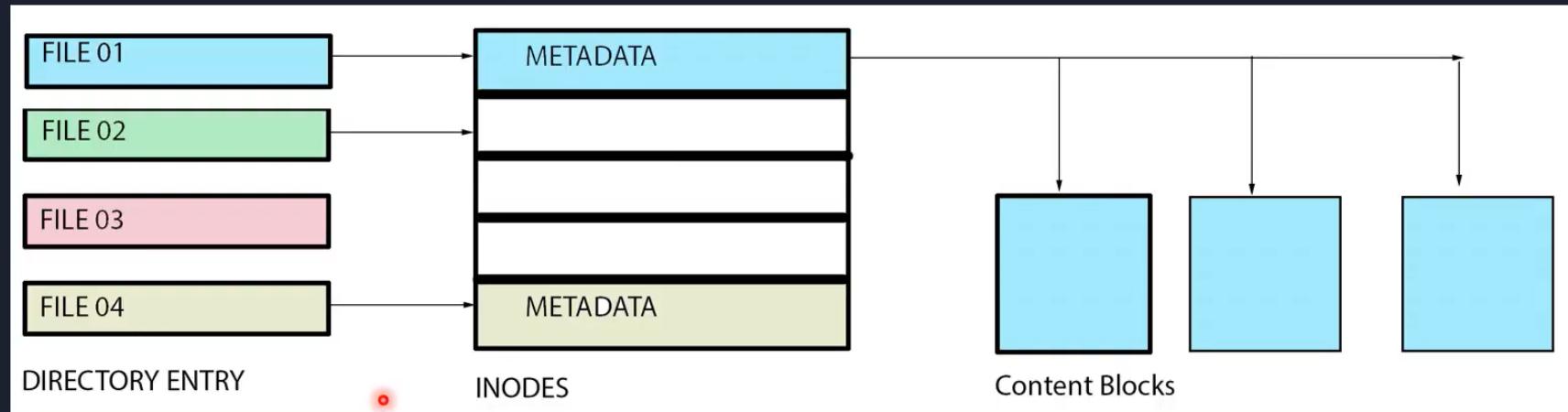
SUPERBLOCK

00000400	00 20 00 00 00 80 00 00-66 06 00 00 2E 74 00 00f....t..
00000410	E9 1F 00 00 01 00 00 00-00 00 00 00 00 00 00 00	é.....
00000420	00 20 00 00 00 20 00 00-00 08 00 00 19 B9 58 54XT
00000430	19 B9 58 54 02 00 FF FF-53 EF 00 00 01 00 00 00	-XT-ÿSi.....
00000440	7F B7 58 54 00 00 00 00-00 00 00 00 01 00 00 00	-XT.....
00000450	00 00 00 00 0B 00 00 00-80 00 00 00 38 00 00 008....
00000460	02 00 00 00 01 00 00 00-7B F3 FF 57 30 12 4D 29{óyW0-M)
00000470	A4 46 E8 C0 12 24 B6 B2-00 00 00 00 00 00 00 00	xFèÀ-\$¶.....
00000480	00 00 00 00 00 00 00 00-2F 6D 65 64 69 61 2F 67/media/g
00000490	65 6E 6F 65 2F 37 62 66-33 66 66 35 37 2D 33 30	enoe/7bf3ff57-30
000004a0	31 32 2D 34 64 32 39 2D-61 34 34 36 2D 65 38 63	12-4d29-a446-e8c
000004b0	30 31 32 32 34 62 36 62-32 00 00 00 00 00 00 00	01224b6b2.....
000004c0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 7F00
000004d0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000004e0	00 00 00 00 00 00 00 00-00 00 00 00 CE FA A4 64Íú¤d
000004f0	93 F1 48 02 B7 41 D3 1E-77 EF 20 3F 01 00 00 00	-ñH-AÓ-wi ?
00000500	0C 00 00 00 00 00 00 00-00 7F B7 58 54 00 00 00 00XT
00000510	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000520	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00000530	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

Cursor pos = 1024; log sec = 2; phy sec = 264194

- 0x00 (4) - Number of inodes in file system
- 0x04 (4) - Number of blocks in file system
- 0x08 (4) - Number of blocks reserved to prevent file system from filling up
- 0x0C (4) - Number of unallocated blocks
- 0x10 (4) - Number of unallocated inodes
- 0x14 (4) - Block where block group 0 starts
- 0x18 (4) - Block size (saved as the number of places to shift 1,024 to the left)
- 0x1C (4) - log cluster size (saved as the number of bits to shift 1,024 to the left)
- 0x20 (4) - Number of blocks in each block group
- 0x24 (4) - Number of fragments in each block group
- 0x28 (4) - Number of inodes in each block group
- 0x2C (4) - Last mount time
- 0x30 (4) - Last written time
- 0x34 (2) Current mount count
- 0x38 (2) - Signature 0xEF53

DIRECTORY ENTRIES



DIRECTORY ENTRIES

Name	Size	Type	Date Modified
Screenshot from 2014-...	141	Regular File	17.12.2014 14:2...
Screenshot from 2014-...	184	Regular File	17.12.2014 14:3...
Screenshot from 2014-...	227	Regular File	17.12.2014 14:4...
Screenshot from 2014-...	91	Regular File	18.12.2014 6:35...

000	56 00 02 00 0C 00 01 02-2E 00 00 00 01 00 02 00	V.....,....,...
010	0C 00 02 02 2E 2E 00 00 C7 01 02 00 30 00 27 01,C...0'.
020	53 63 72 65 65 6E 73 68-6F 74 20 66 72 6F 6D 20	Screenshot from
030	32 30 31 34 2D 31 32 2D-31 37 20 31 36 3A 32 39	2014-12-17 16:29
040	3A 35 37 2E 70 6E 67 00-ED 00 02 00 30 00 27 01	:57.png,i...0'.
050	53 63 72 65 65 6E 73 68-6F 74 20 66 72 6F 6D 20	Screenshot from
060	32 30 31 34 2D 31 32 2D-31 37 20 31 36 3A 33 31	2014-12-17 16:31
070	3A 32 38 2E 70 6E 67 00-E5 00 02 00 30 00 27 01	:28.png,å...0'.
080	53 63 72 65 65 6E 73 68-6F 74 20 66 72 6F 6D 20	Screenshot from
090	32 30 31 34 2D 31 32 2D-31 37 20 31 36 3A 34 30	2014-12-17 16:40
0a0	3A 35 37 2E 70 6E 67 00-BD 00 02 00 58 0F 27 01	:57.png,...X'.
0b0	53 63 72 65 65 6E 73 68-6F 74 20 66 72 6F 6D 20	Screenshot from
0c0	32 30 31 34 2D 31 32 2D-31 38 20 30 38 3A 33 35	2014-12-18 08:35
0d0	3A 31 38 2E 70 6E 67 00-00 00 00 00 00 00 00 00 00	:18.png,.....
0e0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00
0f0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00

Every directory starts with "dot-two dots" directory entries ("." "..") for current and parent directories (analog with FAT subdirectories), followed by file or subfolder entries

DIRECTORY ENTRIES

Name	Size	Type	Date Modified
Screenshot from 2014-...	141	Regular File	17.12.2014 14:2...
Screenshot from 2014-...	184	Regular File	17.12.2014 14:3...
Screenshot from 2014-...	227	Regular File	17.12.2014 14:4...
Screenshot from 2014-...	91	Regular File	18.12.2014 6:35...

000	56 00 02 00 0C 00 01 02-2E 00 00 00 01 00 02 00	V.....
010	0C 00 02 02 2E 2E 00 00 C7 01 02 00 30 00 27 01	C...0..
020	53 63 72 65 65 6E 73 68-6F 74 20 66 72 6F 6D 20	Screenshot from	
030	32 30 31 34 2D 31 32 2D-31 37 20 31 36 3A 32 39	2014-12-17 16:29	
040	3A 35 37 2E 70 6E 67 00-ED 00 02 00 30 00 27 01	:57.png i...0..	
050	53 63 72 65 65 6E 73 68-6F 74 20 66 72 6F 6D 20	Screenshot from	
060	32 30 31 34 2D 31 32 2D-31 37 20 31 36 3A 33 31	2014-12-17 16:31	
070	3A 32 38 2E 70 6E 67 00-E5 00 02 00 30 00 27 01	:28.png å...0..	
080	53 63 72 65 65 6E 73 68-6F 74 20 66 72 6F 6D 20	Screenshot from	
090	32 30 31 34 2D 31 32 2D-31 37 20 31 36 3A 34 30	2014-12-17 16:40	
0a0	3A 35 37 2E 70 6E 67 00-BD 00 02 00 58 0F 27 01	:57.pngX..	
0b0	53 63 72 65 65 6E 73 68-6F 74 20 66 72 6F 6D 20	Screenshot from	
0c0	32 30 31 34 2D 31 32 2D-31 38 20 30 38 3A 33 35	2014-12-18 08:35	
0d0	3A 31 38 2E 70 6E 67 00-00 00 00 00 00 00 00 00	:18.png	
0e0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
0f0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	

0x00 (4) - Inode number
0x04 (2) - directory entry size in bytes
0x06 (1) - name length (b)
0x07 (1) - file type (01 - regular file, 02 - directory, 03 - character device, 04 - block device, 05 - FIFO, 06 - socket, 07- symbolic link)
0x8 (~) - file name in ASCII

(note, that last file name length is 3928 (b) - points to the end of the block)

THE INODES

- In ExtX, a file's primary metadata is stored in an inode structure. Additional metadata can be stored in extended attributes
- Superblock defines the size of an inode (128 or 256 bytes)
- Inodes are counted in the Inode Table starting from 1
- Inodes 1 - 11 are typically reserved, inode 2 is used for the root directory
- Ext2/Ext3 uses “indirect pointers” to track the file data, Ext4 uses “extents”



offset: 0x58 (inode size)

EXT2/EXT3 INODE STRUCTURE

000580	A4 81 00 00	00 90 01 00	74 E6 0C 62	74 E6 0C 62
000590	74 E6 0C 62	00 00 00 00	00 00 00 00	01 00 CA 00 00 00
0005a0	00 00 00 00	01 00 00 00	01 04 00 00	02 04 00 00
0005b0	03 04 00 00	04 04 00 00	05 04 00 00	06 04 00 00
0005c0	07 04 00 00	08 04 00 00	09 04 00 00	0A 04 00 00
0005d0	0B 04 00 00	0C 04 00 00	04 02 00 00	00 00 00 00
0005e0	00 00 00 00	40 CA 05 3D	00 00 00 00	00 00 00 00
0005f0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00

- 0x00 (2) - File mode (ACL)
- 0x4 (4) file size (in bytes)
- 0x8 (4) - last access time (Unix 32 bit, seconds since Jan 1, 1970 UTC)
- 0xC (4) -last inode change time
- 0x10 (4) - last data modification time
- 0x14 (4) - deletion time
- 0x1A (2) - hard link count
- 0x1C (4) - file size in blocks
- 0x28 - start of indirect pointers

EXT4 INODE STRUCTURE

80 81 E8 03 57 0B 00 00-70 93 91 54 67 93 91 54	·è·W··p··Ig··T
67 93 91 54 00 00 00 00-E8 03 01 00 08 00 00 00 00	g··T··è···
00 00 08 00 01 00 00 00 0A F3 01 00 04 00 00 00 006.....
00 00 00 00 00 00 00-01 00 00 00 0A 86 08 00
00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00 00
00 00 00 00 EC F8 63 E9-00 00 00 00 00 00 00 00 00 00	...iscé...
00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00 00
1C 00 00 00 AC 9E A7 4D-AC 9E A7 4D D4 19 75 E1SM...SM...uá
67 93 91 54 AC 9E A7 4D-00 00 00 00 00 00 00 00 00 00	g··T··SM
00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00 00

Extent tree header (12 bytes) with
signature 0xF30A
offset 0x06 (2b) - depth of extent
tree, if =0, then extent node points
to data blocks

Extent node (leaf node, 12bytes)
offset 0x06 (2) - higher 16-bits of
the starting block number
offset 0x08 (4) - lower 32 bits of
the starting block number

0201c600 A4 81 E8 03 B7 31 02 00-F7 98 91 54 66 93 91 54	·x···1···+··Tf··T
0201c610 66 93 91 54 00 00 00-00-E8 03 01 00 20 01 00 00	f··T··è···
0201c620 00 00 08 00 01 00 00 00-00-0A F3 01 00 04 00 00 00ó.....
0201c630 00 00 00 00 00 00 00 00-00-24 00 00 00 D0 81 08 00\$·D.....
0201c640 00 00 00 00 00 00 00 00-00-00 00 00 00 00 00 00 00 00
0201c650 00 00 00 00 00 00 00 00-00-00 00 00 00 00 00 00 00 00
0201c660 00 00 00 00 E8 F8 63 E9-00 00 00 00 00 00 00 00 00 00	...èccé.....
0201c670 00 00 00 00 00 00 00 00-00-00 00 00 00 00 00 00 00 00
0201c680 1C 00 00 00 94 D2 68 00-94 D2 68 00 C8 66 6D D1òh..òh..ÈfmÑ
0201c690 65 93 91 54 C0 97 F5 C2-00 00 00 00 00 00 00 00 00	e··TÀ·ôÂ.....
0201c6a0 00 00 00 00 00 00 00 00-00-00 00 00 00 00 00 00 00 00
0201c6b0 00 00 00 00 00 00 00 00-00-00 00 00 00 00 00 00 00 00
0201c6c0 00 00 00 00 00 00 00 00-00-00 00 00 00 00 00 00 00 00
0201c6d0 00 00 00 00 00 00 00 00-00-00 00 00 00 00 00 00 00 00
0201c6e0 00 00 00 00 00 00 00 00-00-00 00 00 00 00 00 00 00 00
0201c6f0 00 00 00 00 00 00 00 00-00-00 00 00 00 00 00 00 00 00

EXT4 INODE STRUCTURE

06601d00	B4 81 E8 03 00 00 80 0C-C5 0C 16 57 6D 13 16 57	'·è.....·Å·Wm··W	The tree depth is 0x0001
06601d10	31 10 16 57 00 00 00-E8 03 01 00 08 40 06 00	1..W...è....@..	
06601d20	00 00 08 00 01 00 00 00-0A F3 01 00 04 00 01 00ó.....	
06601d30	00 00 00 00 00 00 00-6E 9D 18 00 00 00 3A 00n.....	
06601d40	01 00 00 00 FF 2F 00 00-00 D0 39 00 00 30 00 00ÿ/..·Đ9..·0..	
06601d50	00 20 00 00 A0 39 00-00 50 00 00 00 58 00 00g..·P..·X..	
06601d60	00 00 3A 00 88 39 9C 5C-00 00 00 00 00 00 00 009\.....	
06601d70	00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00	
06601d80	1C 00 00 00 24 23 18 9E-00 00 00 00 00 00 00 00\$#.....	
06601d90	C5 0C 16 57 6C 37 08 66-00 00 00 00 00 00 00 00	·Å·W17·f.....	
06601da0	00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00	
06601db0	00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00	
06601dc0	00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00	
06601dd0	00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00	
06601de0	00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00	
06601df0	00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00	



- extent tree header



- logical block number 0x00000000



- lower bits of physical block address of sub-tree
(0x189E6E = 1613166 block)



- upper bits for block address of the sub-tree