File Systems and FAT File Systems

FAT: File Allocation Table

Nelson Chapter 5, pp 209-212

Carrier, Chapter 9, pp 211-227

Carrier, Chapter 8 (Carrier's Basic Reference Model) not covered but worthwhile.

Some General Comments

A significant technology contribution that Carrier makes is his ***Basic Reference Model*** for file systems

When studying file systems and using TSK, it helps to understand this model

But some file systems cannot be easily mapped into Carrier's model

Notable among these are FAT file systems

Some General Comments

In Chapters 9 & 10 Carrier tried to “force” FAT file systems into his *Basic Reference Model*

In my opinion doing this makes these chapters confusing

Nelson, unfortunately, gives little attention to FAT

About 4 pages

We will try, in the slides, to make FAT easier to understand

But before we look at FAT file systems, there are a few other topics to discuss

File systems generally

Kilobytes…

Clusters

File Systems

Reprise: Disk Formatting

There are three things that must be done to a disk in the order shown

1st: Physical formatting

- Often called "**low level formatting**"
- Done on entire disk

2nd: Partitioning

- Conceptually separates the disk into disjoint non-overlapping parts

3rd. Logical formatting

- Often called "**high level formatting**"
- Done on each partition separately

Logical Formatting and File Systems

Logical formatting involves configuring a file system in a partition

A file system consist of an organization and structure for storing & managing data

Main functions of a file system

- *Tracking free space*
- *Allocating free space*
- *Creating and maintaining directories (i.e., folders)*
- *Creating and maintaining directory and file names*
- *Keeping information as to where the parts of each file are physically stored on the disk*

File Systems

Many different file systems

Different file systems usually work with different OSs

File Systems Commonly Used by Windows

FAT12	(early DOS, diskettes)
FAT16	(later DOS, all Windows OSs)
FAT32	(All Windows OSs that I know of)
exFat	(Win Vista and beyond)
VFAT	(FAT32 with long names (Used in Win95, 98, ME)
FATX	(Xbox (But can be read by any current Win OS)
NTFS	(windows NT, 2000, XP, 2003, win7…win11)

File Systems Commonly Used by Unix, Linux & MacOSs

HPFS	(OS/2, Windows NT)
NWFS	(NetWare)
Ext2, Ext3, Ext4	(Linux)
UFS1, UFS2, ZFS	(SunOS)
HFS	(MacOS up to 8)
HFS+	(MacOS 8.1 and later)
Reiser	(Linux, SUSI Linux)
Others as well	

Kilobytes…

How Big?

How big is a Kilobyte?

$10^3 = 1{,}000$ *bytes ?*

$2^{10} = 1{,}024$ *bytes ?*

How big is a megabyte?

$10^6 = (10^3)^2 = 1{,}000{,}000$ *bytes ?*

$2^{20} = (2^{10})^2 = 1024^2 = 1{,}048{,}576$ *bytes ?*

$10^3 \text{ x } 2^{10} = 1000 \text{ x } 1024 = 1{,}024{,}000$ *bytes ?*

What about a gigabyte?

$10^9 = (10^3)^3 = 1{,}000{,}000{,}000$ *bytes ?*

$2^{30} = (2^{10})^3 = 1024^3 = 1{,}073{,}741{,}824$ *bytes ?*

$10^6 \text{ x } 2^{10} = 10^6 \text{ x } 1024 = 1{,}024{,}000{,}000$ *bytes ?*

Context

The values used depend on context

If were speaking of transmission bit rates, the conventional mathematical values are used

Kilobyte/sec. = 10^3 = 1000 bytes/sec.

Megabyte/sec. = 10^6 = 1,000,000 bytes/sec.

Gigabyte/sec. = 10^9 = 1,000,000,000 bytes/sec.

If were discussing computer storage, then the power of two numbers are usually used

Kilobyte = 2^{10} = 1024 bytes

Megabyte = 2^{20} = 1,048,576 bytes

Gigabyte = 2^{30} = 1,073,741,824 bytes

But Not Always

Often the capacity of removable storage is expressed by mixing the two systems

A 1 gigabyte flash drive could be 10^6 kilobytes

10^6 x 1024 = 1,024,000,000 bytes

Example

A Cruzer 1GB flash drive

Marketed as a 1GByte flash drive

1,002,438,144 byes in 1,957,887 sectors

A flash drive used in RADISH

Marketed as a 32 MByte flash drive

29,696,000 bytes in 58,000 sectors

These don't fit either system

How long would it take?

Suppose that you want to transmit a 1GB file over a 1MB link

How long would it take?

10^9 Bytes / 10^6 Bytes per second = 1000 seconds?

NO!

2^{30} bytes / 10^6 bytes/sec. = 1,073,741,824 / 1,000,000

= 1,073 seconds

Clusters

Clusters

All data on a FAT, NTFS, or HPFS partition is stored in allocation units called ***clusters***

Cluster are the smallest unit of disk capacity that can be allocated or deallocated by the file system

- *It is atomic for the file system*

Clusters consist of fixed number of contiguous sectors on a disk

For some file systems cluster size is determined by the size of the partition

- *The larger the partition, the larger the cluster size*
- *FAT is such a file system*

Clusters

If a cluster size is 32 sectors (i.e, 16 KB), even a 1 KB file will consume the entire cluster

Using large cluster sizes leaves a lot of unused space

Comments

- *Much space is wasted*
- *There's ample places for stuff to hide*

FAT File Systems

File **A**llocation **T**able File Systems

FAT File Systems
Overview

All FAT file systems have three parts

- *A reserved area*
- *The File Allocation Tables area*
- *The data area*
  - Root directory
  - Content area

The term “**FAT**” can refer to either

- *The entire FAT file system (FATfs) or*
- *The File Allocation Tables (FATat) area*
- *This sometimes causes confusion*

FAT File Systems *(FATfs)*

Versions

There are several versions of FATfs, some of which are

FAT12

FAT16

FAT32

exFAT (or FATex)

VFAT

FATX

FATfs are widely used in flash devices of all types

FAT

Where are FAT file systems used?

- *Floppy diskettes (Remember these?)*
- *USB “thumb" drives*
- *Compact "flash" cards*
- *Camera & cell phone SD memory*
- …

Modern OSs use other file systems

- *But all OSs that I know of also read and write FAT file systems*

FAT12

FAT12 is used specifically on floppy disks

Limited amount of file space

Originally designed for MS-DOS 1.0

- *First Microsoft OS*
- *Used only floppy disks – not on hard disks*

Still used for floppies today – if floppies are used

Cluster size is one sector

FAT16

Used on OSs such as

- *MS-DOS 3.0 and later*
- *Windows95, release 1*
- *NT 3.5 and 4.0*
- *USB “thumb” drives*

Partition sizes limited to ≤ 2.02 GB

Cluster size varies with the size of the disk

FAT32

Used with

- *Windows 95, release 2, Windows 98 and ME*
- *Windows 2000 – Win11*
- *USB “thumb” drives*

Usually only be formatted up to 32 GB with Windows

- *Can be formatted beyond this size but cluster sizes are very large*

Number of directory entries ≤ 65,536

exFAT

Becoming widely used for very large flash drives and flash memory; e.g., videos…

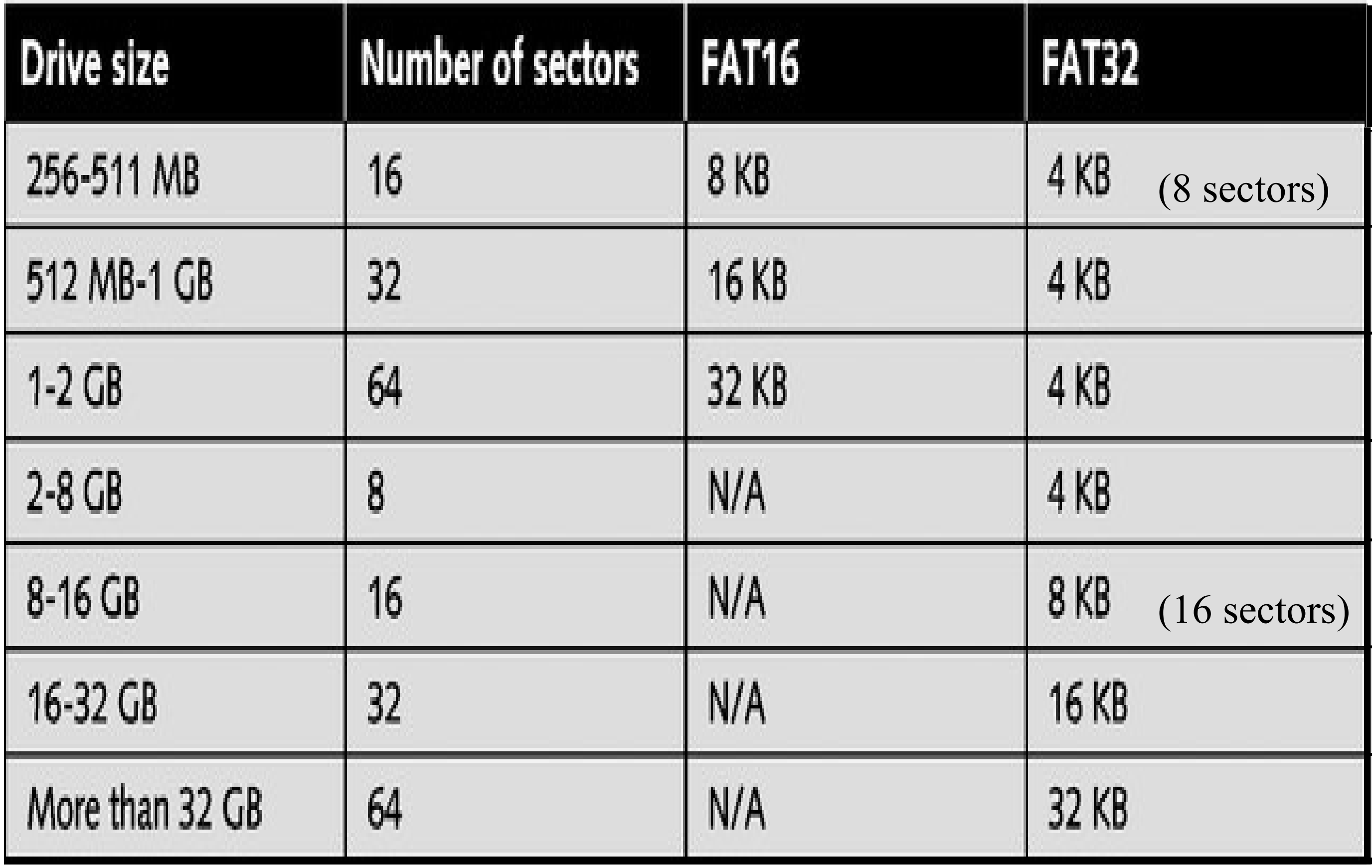
Capacity up to 512 TB

Number of directory entries ≤ 270 million

Cluster sizes

7 MB–256 MB	*4 KB*
256 MB–32 GB	*32 KB*
32 GB–256 TB	*128 KB*
> 256 TB	*Not supported*

FAT16, FAT32 & exFAT Cluster Size Comparison



Drive size	Number of sectors	FAT16	FAT32	exFAT
256-511 MB	16	8 KB	4 KB (8 sectors)	4KB up to 256MB 32KB, 257MB-511MB
512 MB-1 GB	32	16 KB	4 KB	32 KB
1-2 GB	64	32 KB	4 KB	32 KB
2-8 GB	8	N/A	4 KB	32 KB
8-16 GB	16	N/A	8 KB (16 sectors)	32 KB
16-32 GB	32	N/A	16 KB	32 KB
More than 32 GB	64	N/A	32 KB	128KB up to 256TB >256TB not supported

Lots of places for stuff to hide

FAT Volume Organization

A FAT file system volume has the following components in order

Carrier's terms

Partition Boot Sector(s) (PBS) ← Reserved Area

File Allocation Table (FAT1)

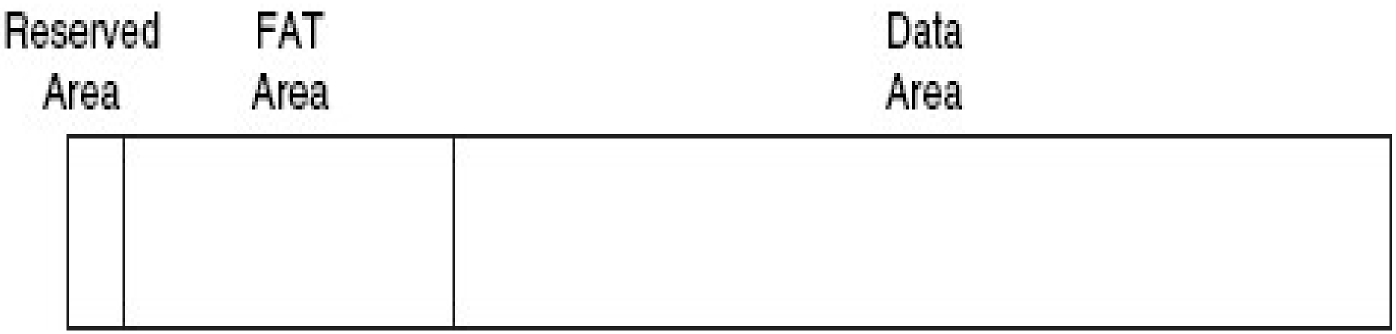
Mirror of FAT table (FAT2)

FAT Area

Root Folder (or Directory)

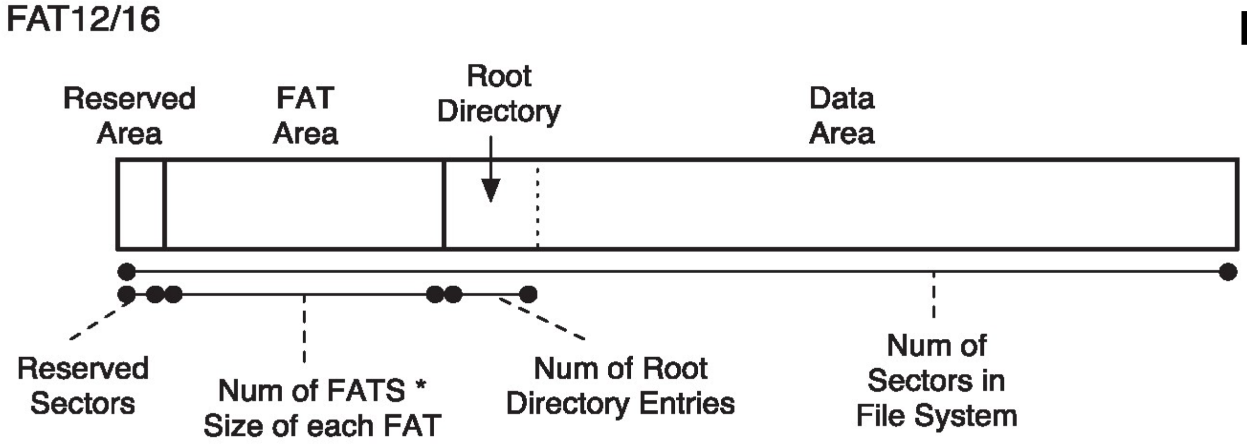
Content of other Folders and files

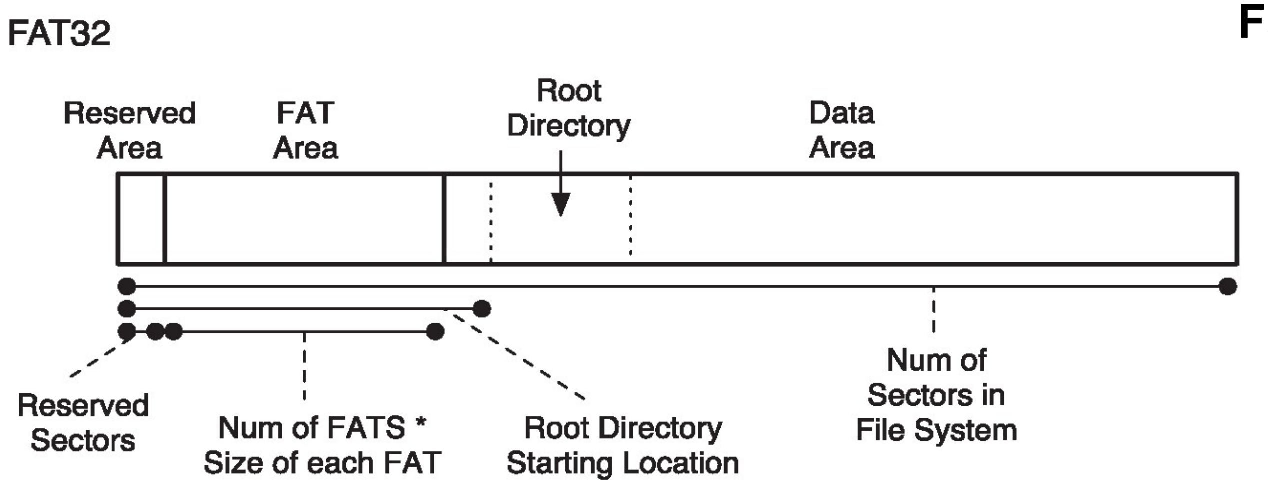
Data Area



Some Differences Between FAT12/16 and FAT32

FAT32 has somewhat different layout than FAT12 & FAT16


FAT12/16
Reserved Area
FAT Area
Root Directory
Data Area
Reserved Sectors
Num of FATS * Size of each FAT
Num of Root Directory Entries
Num of Sectors in File System


FAT32
Reserved Area
FAT Area
Root Directory
Data Area
Reserved Sectors
Num of FATS * Size of each FAT
Root Directory Starting Location
Num of Sectors in File System

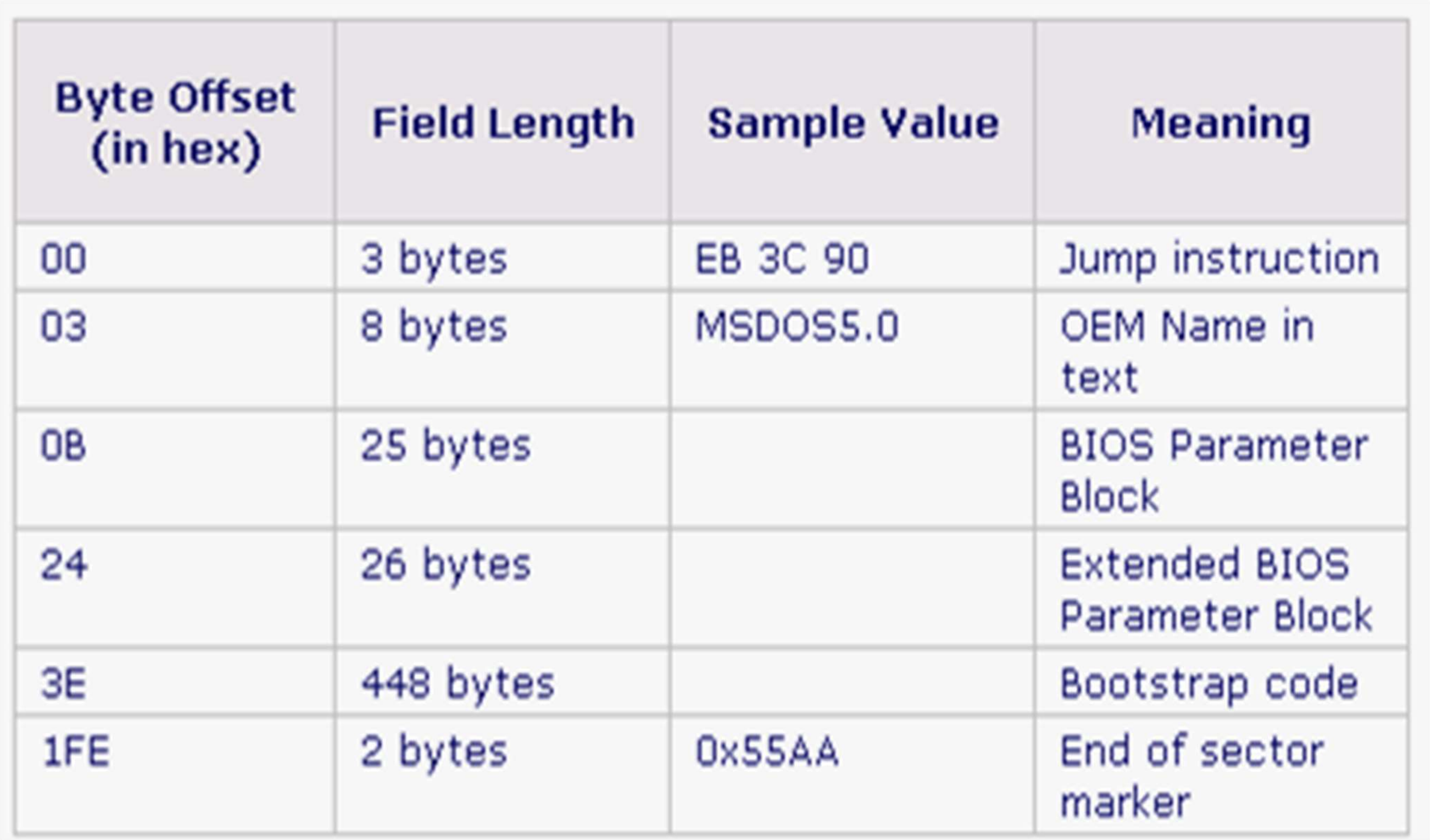
FAT12 &16

- *One sector in Reserved Area (i.e., PBS)*
  - *Sometimes > 1 for flash drives*
- *Root Directory*
  - Fixed size defined in PBS
  - Max of 512 entries
  - Located immediately after FAT Area

Fat32

- *Can have multiple sectors in Reserved Area (PBS)*
- *Root Directory*
  - Size dynamically defined
  - Location defined in PBS
  - Can be located anywhere in Data Area
  - But usually immediately after FAT area

FAT PBS Format



Byte Offset (in hex)	Field Length	Sample Value	Meaning
00	3 bytes	EB 3C 90	Jump instruction
03	8 bytes	MSDOS5.0	OEM Name in text
0B	25 bytes		BIOS Parameter Block
24	26 bytes		Extended BIOS Parameter Block
3E	448 bytes		Bootstrap code
1FE	2 bytes	0x55AA	End of sector marker

FATat
The File Allocation Table Itself

The ***FATfs*** is divided into a fixed number of clusters, each of the same number of sectors

All clusters in a specific **FATfs** *are the same size*

Files in the *data area* (such as user files) are contained in one or more clusters that are not necessarily contiguous

The ***FATat*** (FAT allocation table) is a map of all the clusters in the ***FATfs***

List of entries, one for each cluster in the FATfs

FATat
*The File Allocation Table Itself**

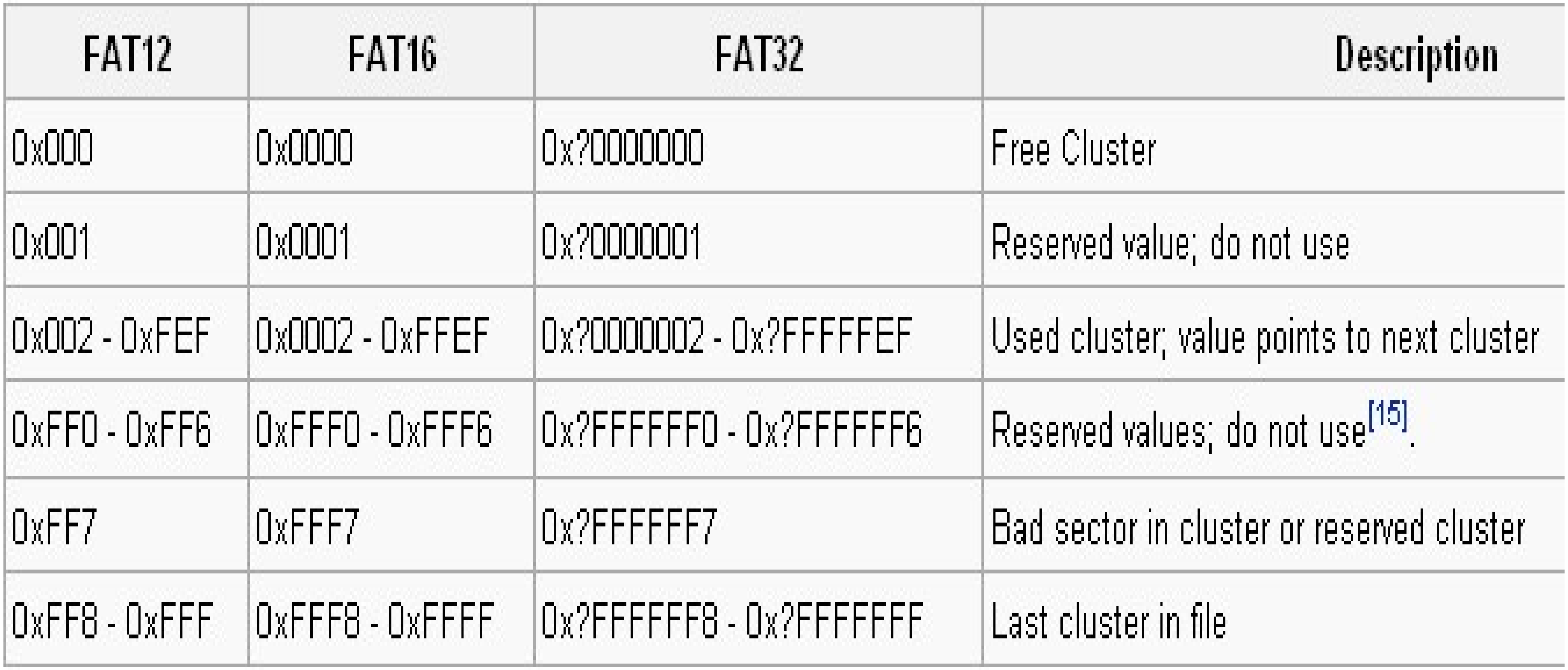
The ***FATat*** contains, for each cluster, one of the following

- *The cluster number of the next cluster in a chain*
- *An entry indicating the end of cluster chain (EOC)*
  - i.e., the last cluster in a file or folder
- *An entry to mark a bad cluster*
- *An entry to mark a reserved cluster*
- *An entry to note that the cluster is unused*

The number of bits per ***FATat*** entry is

FAT12:	*12 bits*
FAT16:	*16 bits (2 bytes)*
FAT32:	*32 bits (uses 28 bits)*
exFAT:	*32 bits*

Contents of FATat Entries
But Not exFAT

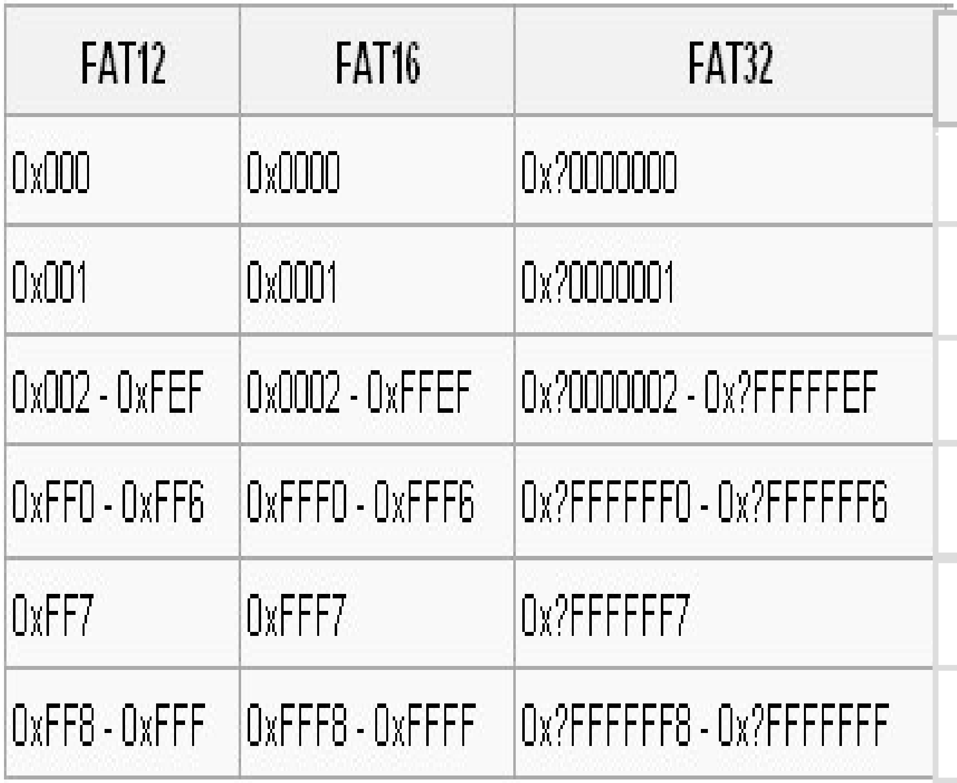


FAT12	FAT16	FAT32	Description
0x000	0x0000	0x?0000000	Free Cluster
0x001	0x0001	0x?0000001	Reserved value; do not use
0x002 - 0xFEF	0x0002 - 0xFFEF	0x?0000002 - 0x?FFFFFEF	Used cluster; value points to next cluster
0xFF0 - 0xFF6	0xFFF0 - 0xFFF6	0x?FFFFFF0 - 0x?FFFFFF6	Reserved values; do not use[15].
0xFF7	0xFFF7	0x?FFFFFF7	Bad sector in cluster or reserved cluster
0xFF8 - 0xFFF	0xFFF8 - 0xFFFF	0x?FFFFFF8 - 0x?FFFFFFF	Last cluster in file

The “?” indicates that these 4 bits may be anything.

FAT32 uses only 28 of the 32 bits.

Contents of exFATat Entries*



FAT12	FAT16	FAT32	exFAT	Description
0x000	0x0000	0x?0000000	Not used to indicate free clusters	Free cluster
0x001	0x0001	0x?0000001		Reserved value; do not use
0x002 - 0xFEF	0x0002 - 0xFFEF	0x?0000002 - 0x?FFFFFEF		Used cluster; value points to next cluster
0xFF0 - 0xFF6	0xFFF0 - 0xFFF6	0x?FFFFFF0 - 0x?FFFFFF6		Reserved value; do not use
0xFF7	0xFFF7	0x?FFFFFF7		Bad sector in cluster or reserved cluster
0xFF8 - 0xFFF	0xFFF8 - 0xFFFF	0x?FFFFFF8 - 0x?FFFFFFF		Last cluster in a file

The “?” indicates that these 4 bits may be anything.

FAT32 uses only 28 of the 32 bits.

FAT File/Folder Information Structure

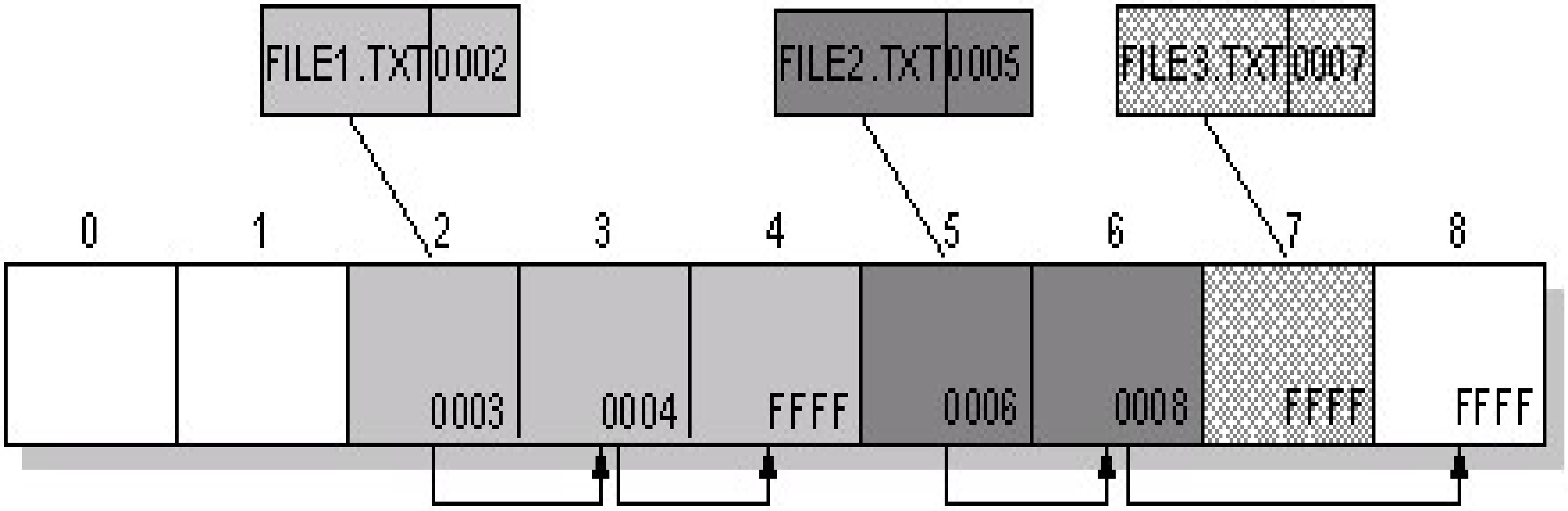
File or folder content is located in clusters pointed to by the “starting cluster number” entry in the FAT

New files are given the first available cluster location on the volume

NT-based OSs can store additional time stamp info

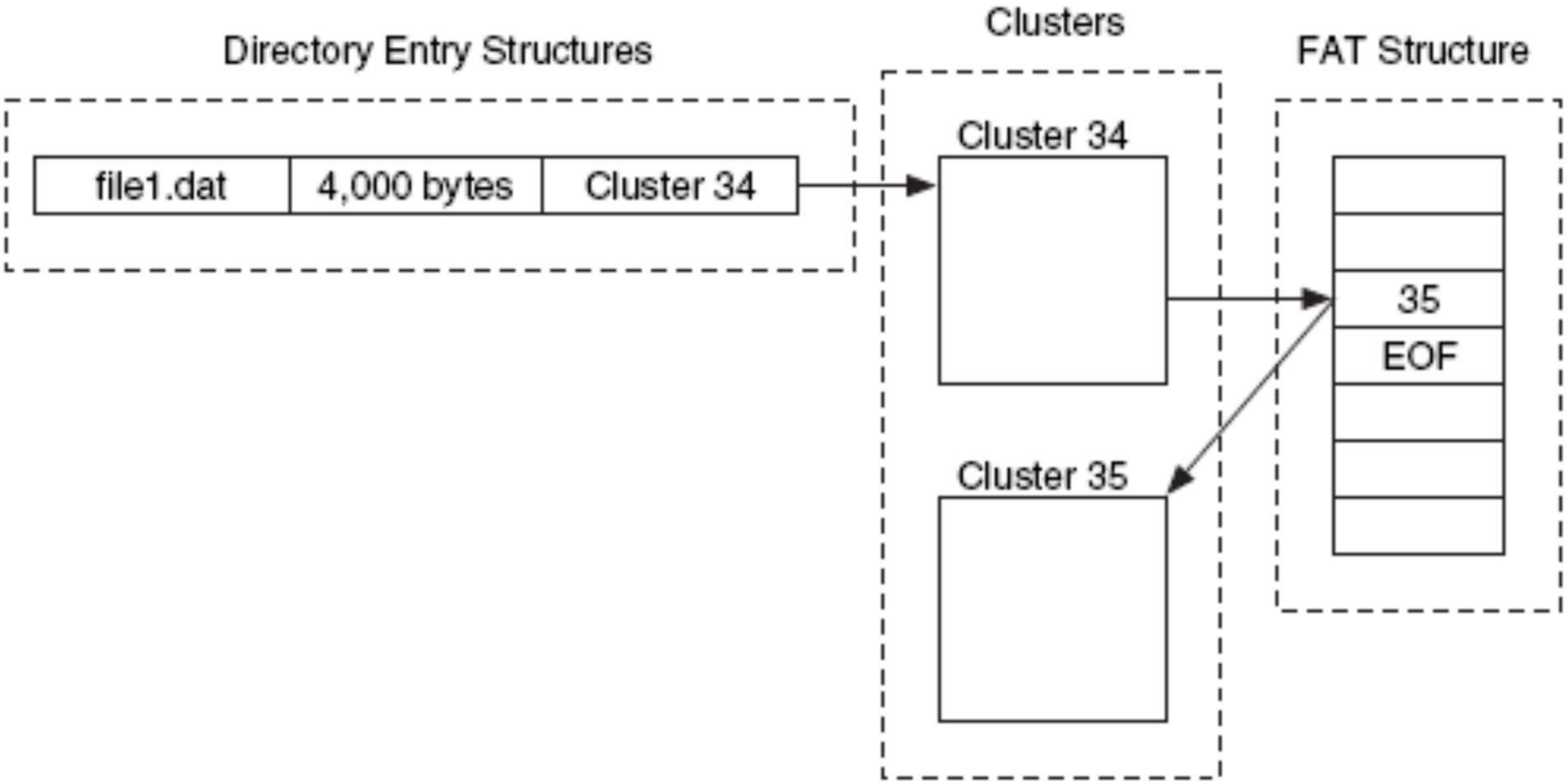
FAT12 & FAT16 Operation

Example: FATat entries for three text files



Directory/FAT Operation

Example: FATfs ***Root Directory*** entry



Root Directory

The ***Root Directory*** has a 32-byte entry for each file, folder, subfolder etc. on the FAT partition

- *File or Folder Name (eight-plus-three characters)*
- *Attribute byte (1 byte)*
  - 1 bit each for folder, file, volume label, archive, system, hidden & read-only
- *Create time (3 bytes)*
- *Create date (2 bytes)*
- *Last access date (2 bytes)*
- *Last modified time (2 bytes)*
- *Last modified date (2 bytes)*
- *Starting cluster number in the file allocation table (2 bytes)*
- *File size (4 bytes)* [4.2 GB. But Nelson writes: 2 GB]

Size of File Name

The space for the file or folder name is 8 characters plus a 3 character extension

The file named **forensic.txt** *just fits*

The file name **profnelson.txt** *doesn't quite fit*

*And what do we do about .doc***x**

Originally FATfs' had names limited to 8+3 characters

But the file system was modified to allow for much larger names

How?

File Names Longer Than 8+3 Characters

A 2nd *Directory* entry located before the standard *Directory* entry for a file or folder

32 bytes just like normal entry

Can contain up to 26 ASCII or 13 UTF-16 characters

These entries can be chained together to allow for very long names

After the last UTF-16 character a 0x00 0x00 is appended

Not used characters are filled with 0xFF

2nd Directory Entry for Long File Names

This 2nd ***Directory*** entry format

File or Folder name (11 bytes) **Part of long name: 11** bytes

Attribute byte (1 byte)

1 bit each for subfolder, file, volume label, archive, system, hidden, read-only

0x0F (volumeLabel, system, hidden, read-only bits all = 1)

Causes MSDOS to ignore the entry

~~Create time~~ (3 bytes) **Part of long name: 03** bytes

~~Create date~~ (2 bytes) **Part of long name: 03** bytes

~~Last access date~~ (2 bytes) **Part of long name: 02** bytes

~~Last modified time~~ (2 bytes) **Part of long name: 02** bytes

~~Last modified date~~ (2 bytes) **Part of long name: 02** bytes

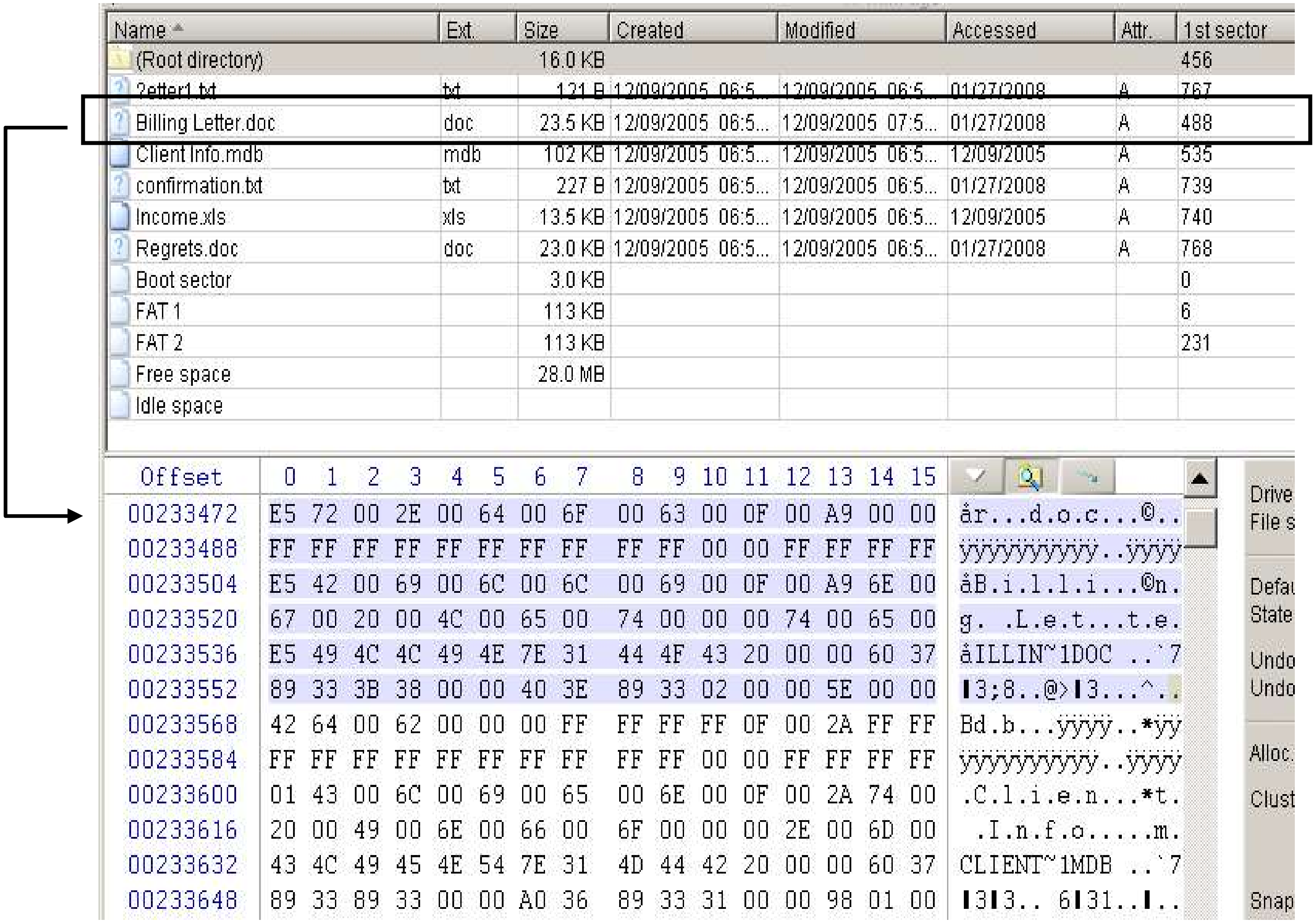
Starting cluster number set to 0x00

~~File size~~ (4 bytes) **Part of long name: 04** bytes

The bytes used to store a long file name total…..………**: 26** bytes

Analysis of a Root Directory Entry

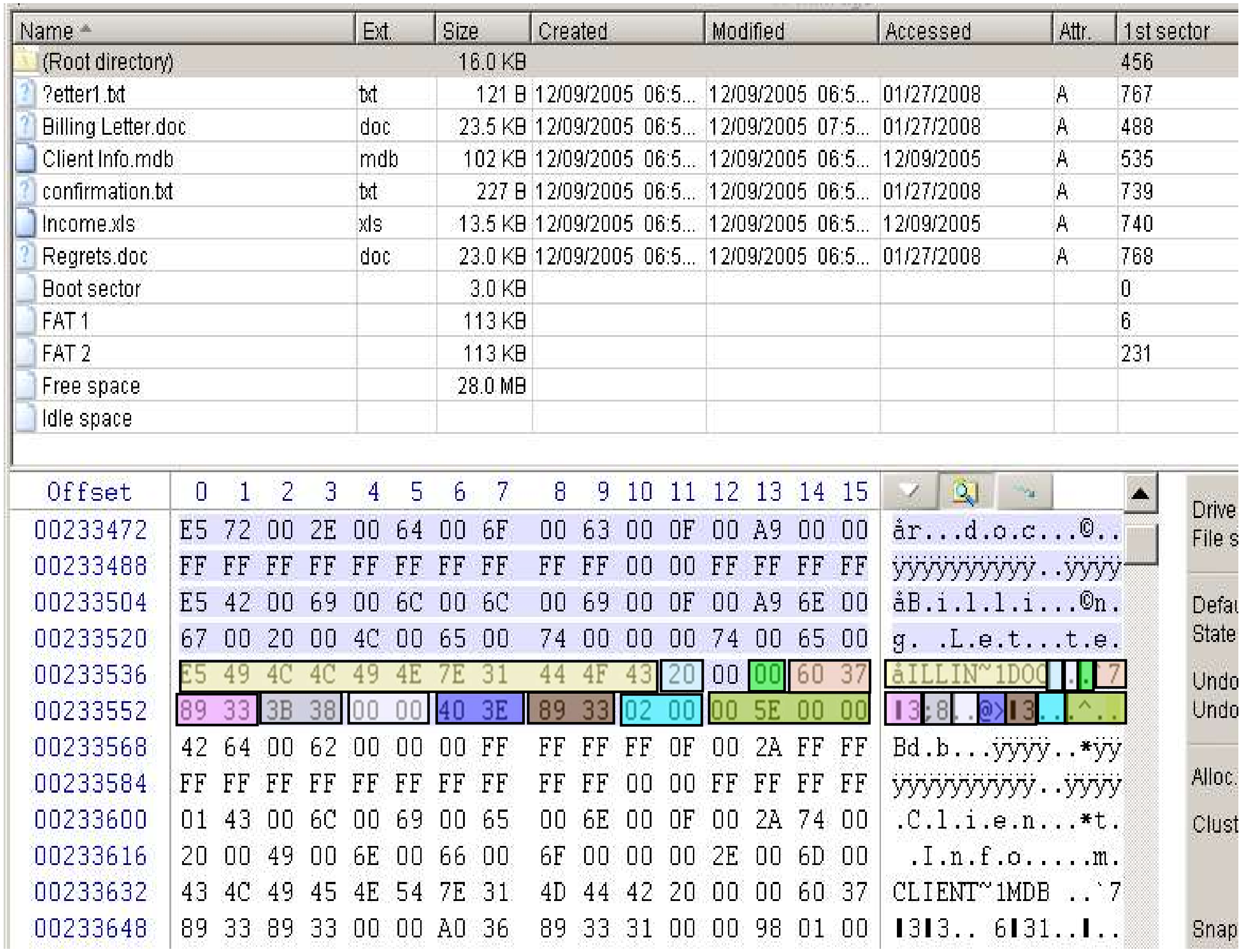
Beginning of Root Directory



Name	Ext.	Size	Created	Modified	Accessed	Attr.	1st sector
(Root directory)		16.0 KB					456
?etter1.txt	txt	121 B	12/09/2005 06:5...	12/09/2005 06:5...	01/27/2008	A	767
Billing Letter.doc	doc	23.5 KB	12/09/2005 06:5...	12/09/2005 07:5...	01/27/2008	A	488
Client Info.mdb	mdb	102 KB	12/09/2005 06:5...	12/09/2005 06:5...	12/09/2005	A	535
confirmation.txt	txt	227 B	12/09/2005 06:5...	12/09/2005 06:5...	01/27/2008	A	739
Income.xls	xls	13.5 KB	12/09/2005 06:5...	12/09/2005 06:5...	12/09/2005	A	740
Regrets.doc	doc	23.0 KB	12/09/2005 06:5...	12/09/2005 06:5...	01/27/2008	A	768
Boot sector		3.0 KB					0
FAT 1		113 KB					6
FAT 2		113 KB					231
Free space		28.0 MB					
Idle space							

```
  Offset    0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15
00233472   E5 72 00 2E 00 64 00 6F  00 63 00 0F 00 A9 00 00   år...d.o.c...©..
00233488   FF FF FF FF FF FF FF FF  FF FF 00 00 FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿ..ÿÿÿÿ
00233504   E5 42 00 69 00 6C 00 6C  00 69 00 0F 00 A9 6E 00   åB.i.l.l.i...©n.
00233520   67 00 20 00 4C 00 65 00  74 00 00 00 74 00 65 00   g. .L.e.t...t.e.
00233536   E5 49 4C 4C 49 4E 7E 31  44 4F 43 20 00 00 60 37   åILLIN~1DOC ..`7
00233552   89 33 3B 38 00 00 40 3E  89 33 02 00 00 5E 00 00   ▮3;8..@>▮3...^..
00233568   42 64 00 62 00 00 00 FF  FF FF FF 0F 00 2A FF FF   Bd.b...ÿÿÿÿ..*ÿÿ
00233584   FF FF FF FF FF FF FF FF  FF FF 00 00 FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿ..ÿÿÿÿ
00233600   01 43 00 6C 00 69 00 65  00 6E 00 0F 00 2A 74 00   .C.l.i.e.n...*t.
00233616   20 00 49 00 6E 00 66 00  6F 00 00 00 2E 00 6D 00    .I.n.f.o.....m.
00233632   43 4C 49 45 4E 54 7E 31  4D 44 42 20 00 00 60 37   CLIENT~1MDB ..`7
00233648   89 33 89 33 00 00 A0 36  89 33 31 00 00 98 01 00   ▮3▮3.. 6▮31..▮..
```

File Names Longer than 8+3
"Billing Letter.doc"
(14+3)



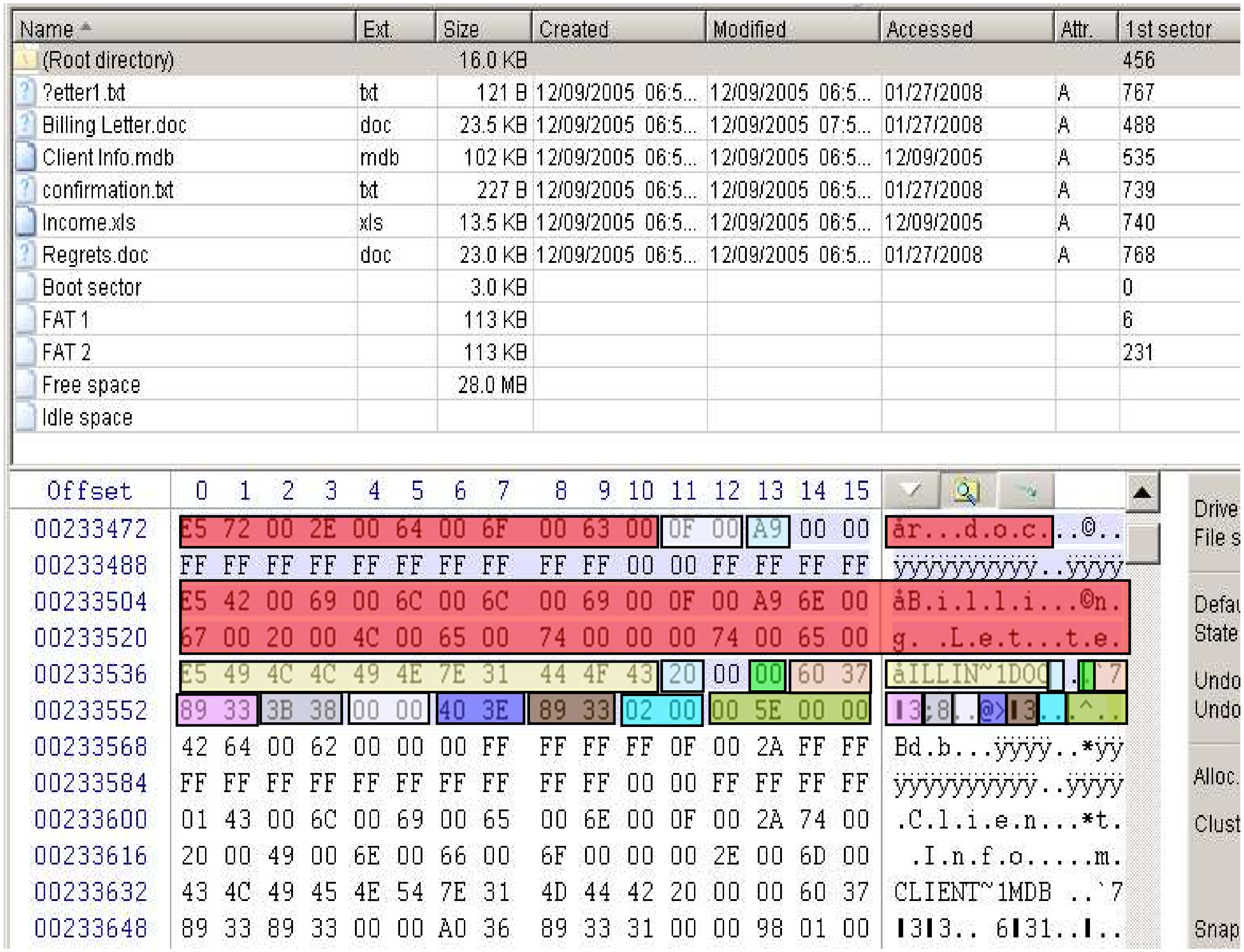
Name	Ext.	Size	Created	Modified	Accessed	Attr.	1st sector
(Root directory)		16.0 KB					456
?etter1.txt	txt	121 B	12/09/2005 06:5...	12/09/2005 06:5...	01/27/2008	A	767
Billing Letter.doc	doc	23.5 KB	12/09/2005 06:5...	12/09/2005 07:5...	01/27/2008	A	488
Client Info.mdb	mdb	102 KB	12/09/2005 06:5...	12/09/2005 06:5...	12/09/2005	A	535
confirmation.txt	txt	227 B	12/09/2005 06:5...	12/09/2005 06:5...	01/27/2008	A	739
Income.xls	xls	13.5 KB	12/09/2005 06:5...	12/09/2005 06:5...	12/09/2005	A	740
Regrets.doc	doc	23.0 KB	12/09/2005 06:5...	12/09/2005 06:5...	01/27/2008	A	768
Boot sector		3.0 KB					0
FAT 1		113 KB					6
FAT 2		113 KB					231
Free space		28.0 MB					
Idle space							

```
  Offset    0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15
00233472   E5 72 00 2E 00 64 00 6F  00 63 00 0F 00 A9 00 00   år...d.o.c...©..
00233488   FF FF FF FF FF FF FF FF  FF FF 00 00 FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿ..ÿÿÿÿ
00233504   E5 42 00 69 00 6C 00 6C  00 69 00 0F 00 A9 6E 00   åB.i.l.l.i...©n.
00233520   67 00 20 00 4C 00 65 00  74 00 00 00 74 00 65 00   g. .L.e.t...t.e.
00233536   E5 49 4C 4C 49 4E 7E 31  44 4F 43 20 00 00 60 37   åILLIN~1DOC ..`7
00233552   89 33 3B 38 00 00 40 3E  89 33 02 00 00 5E 00 00   ▮3;8..@>▮3...^..
00233568   42 64 00 62 00 00 00 FF  FF FF FF 0F 00 2A FF FF   Bd.b...ÿÿÿÿ..*ÿÿ
00233584   FF FF FF FF FF FF FF FF  FF FF 00 00 FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿ..ÿÿÿÿ
00233600   01 43 00 6C 00 69 00 65  00 6E 00 0F 00 2A 74 00   .C.l.i.e.n...*t.
00233616   20 00 49 00 6E 00 66 00  6F 00 00 00 2E 00 6D 00    .I.n.f.o.....m.
00233632   43 4C 49 45 4E 54 7E 31  4D 44 42 20 00 00 60 37   CLIENT~1MDB ..`7
00233648   89 33 89 33 00 00 A0 36  89 33 31 00 00 98 01 00   ▮3▮3.. 6▮31..▮..
```

- 🟨 DOS file name
- ▢ Attribute (archive)
- □ Reserved
- 🟩 Create time (10ms units)
- 🟧 Create time*
- 🟪 Create date*
- ▨ Last access date
- □ Used by OS/2
- 🟦 Last modified time*
- 🟫 Last modified date*
- ▢ First cluster
- 🟩 File size (bytes)

* Bit encoded

File Names Longer than 8+3
"Billing Letter.doc"
(14+3)



Name	Ext.	Size	Created	Modified	Accessed	Attr.	1st sector
(Root directory)		16.0 KB					456
?etter1.txt	txt	121 B	12/09/2005 06:5...	12/09/2005 06:5...	01/27/2008	A	767
Billing Letter.doc	doc	23.5 KB	12/09/2005 06:5...	12/09/2005 07:5...	01/27/2008	A	488
Client Info.mdb	mdb	102 KB	12/09/2005 06:5...	12/09/2005 06:5...	12/09/2005	A	535
confirmation.txt	txt	227 B	12/09/2005 06:5...	12/09/2005 06:5...	01/27/2008	A	739
Income.xls	xls	13.5 KB	12/09/2005 06:5...	12/09/2005 06:5...	12/09/2005	A	740
Regrets.doc	doc	23.0 KB	12/09/2005 06:5...	12/09/2005 06:5...	01/27/2008	A	768
Boot sector		3.0 KB					0
FAT 1		113 KB					6
FAT 2		113 KB					231
Free space		28.0 MB					
Idle space							

```
  Offset      0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15
00233472     E5 72 00 2E 00 64 00 6F  00 63 00 0F 00 A9 00 00   år...d.o.c...©..
00233488     FF FF FF FF FF FF FF FF  FF FF 00 00 FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿ..ÿÿÿÿ
00233504     E5 42 00 69 00 6C 00 6C  00 69 00 0F 00 A9 6E 00   åB.i.l.l.i...©n.
00233520     67 00 20 00 4C 00 65 00  74 00 00 00 74 00 65 00   g. .L.e.t...t.e.
00233536     E5 49 4C 4C 49 4E 7E 31  44 4F 43 20 00 00 60 37   åILLIN~1DOC ..`7
00233552     89 33 3B 38 00 00 40 3E  89 33 02 00 00 5E 00 00   ▌3;8..@>▌3...^..
00233568     42 64 00 62 00 00 00 FF  FF FF FF 0F 00 2A FF FF   Bd.b...ÿÿÿÿ..*ÿÿ
00233584     FF FF FF FF FF FF FF FF  FF FF 00 00 FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿ..ÿÿÿÿ
00233600     01 43 00 6C 00 69 00 65  00 6E 00 0F 00 2A 74 00   .C.l.i.e.n...*t.
00233616     20 00 49 00 6E 00 66 00  6F 00 00 00 2E 00 6D 00    .I.n.f.o.....m.
00233632     43 4C 49 45 4E 54 7E 31  4D 44 42 20 00 00 60 37   CLIENT~1MDB ..`7
00233648     89 33 89 33 00 00 A0 36  89 33 31 00 00 98 01 00   ▌3▌3.. 6▌31..▌..
```

- 🟨 DOS file name
- 🟦 Attribute (archive)
- □ Reserved
- 🟩 Create time (10ms units)
- 🟧 Create time*
- 🟪 Create date*
- ⬜ Last access date
- □ Used by OS/2
- 🟦 Last modified time*
- 🟫 Last modified date*
- 🟦 First cluster
- 🟩 File size (bytes)
- 🟥 Long file name
- □ Reserved
- 🟦 DOS file name checksum

0x00 indicates end of long file name
Unused characters filled with 0xFF

* Bit encoded

Slack and Chaining

Slack

Slack is the unused space in a cluster not used by the file

There are 2 types of slack

- *RAM slack*
- *File slack*

RAM Slack

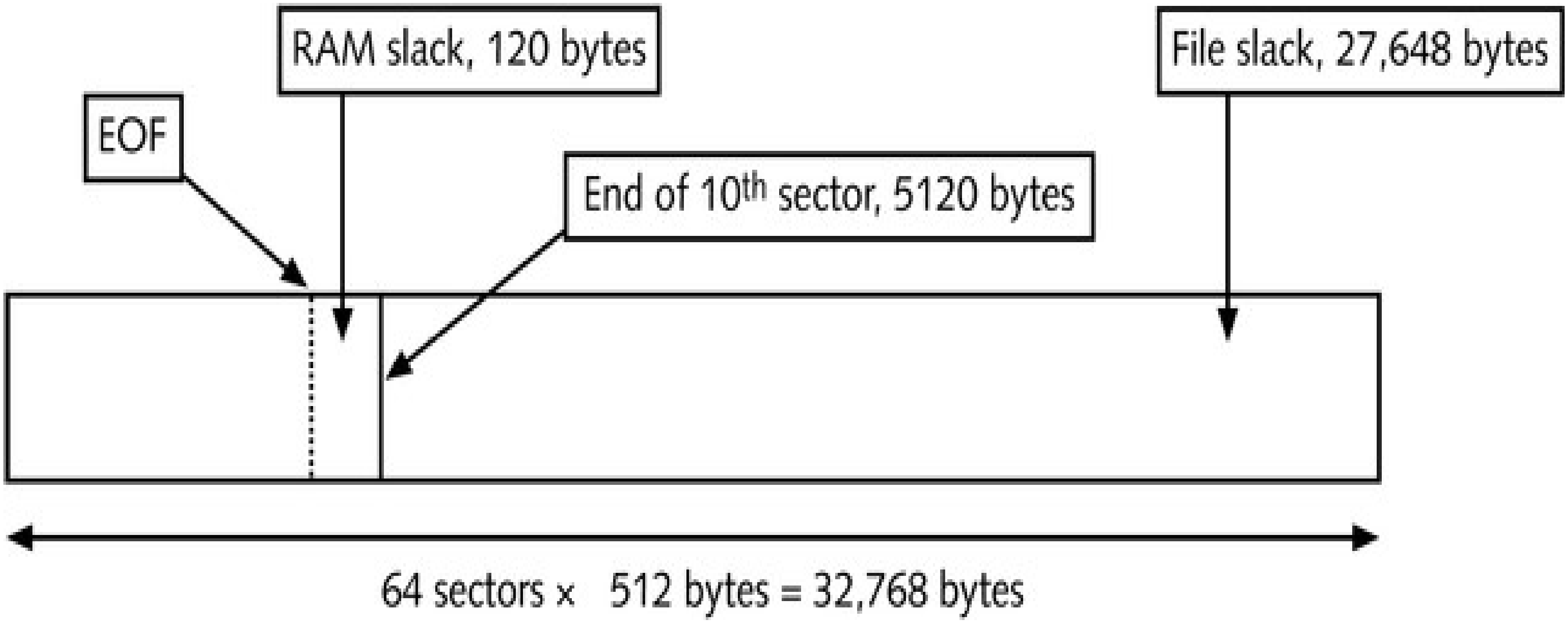
- *The unused space on the last sector of a file*

File Slack (sometimes called Drive Slack)

- *The unused sectors in the cluster in which the file resides*

Slack Example

5000 Byte File on a 40 GB FAT Disk


RAM slack, 120 bytes
File slack, 27,648 bytes
EOF
End of 10th sector, 5120 bytes
64 sectors × 512 bytes = 32,768 bytes

A 5000 byte file occupies 5000/512 or 9.765 sectors

Files Larger Than One Cluster

If a file exceeds the size of a cluster, a 2nd cluster is "chained" to the first cluster

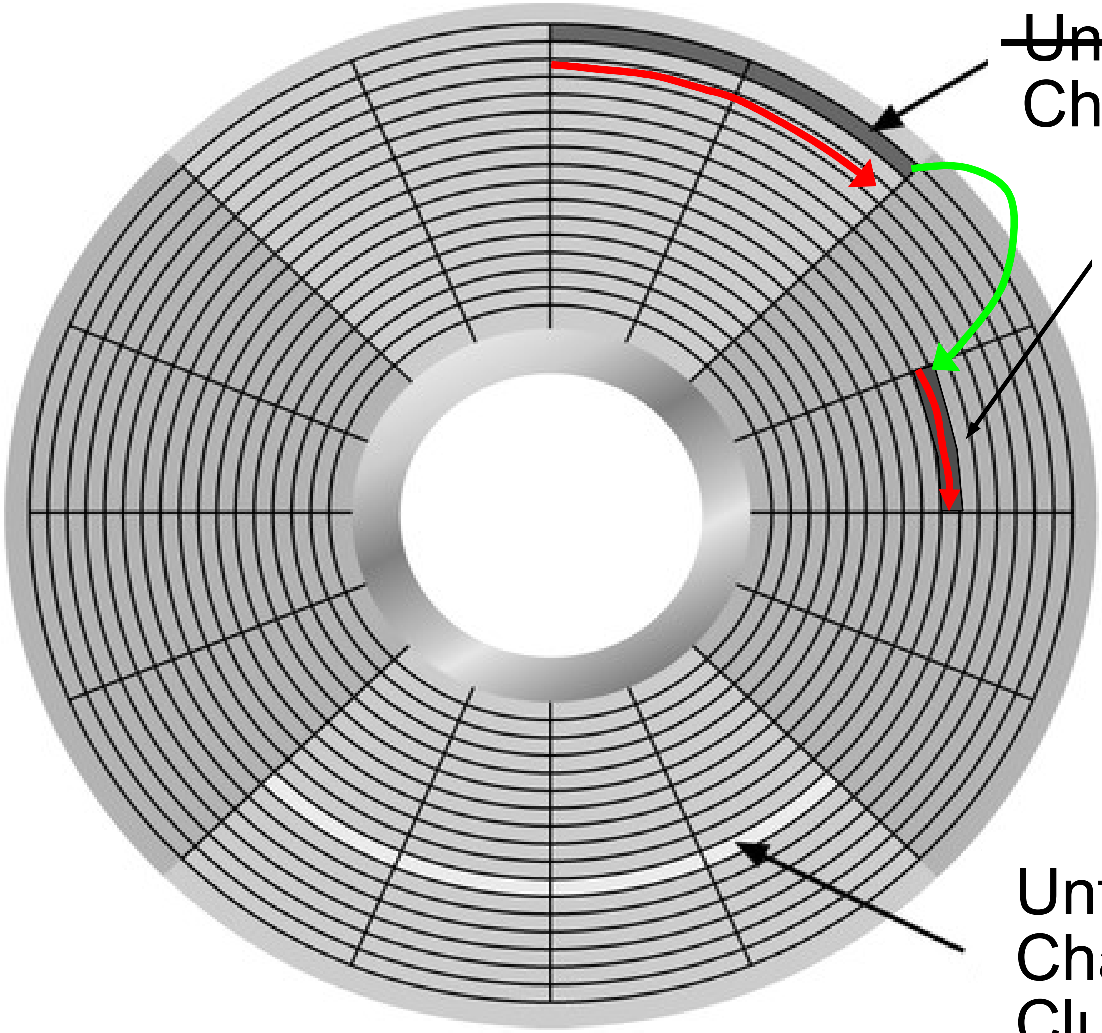
Often these chained clusters are adjacent to each other

But at other times the clusters containing a file are not contiguous

*This is called **fragmentation***

When you defrag a disk, you are rearranging the contents of the disk so that there is minimum fragmentation

Cluster Chaining



~~Un~~fragmented Chained File

Fragment

Unfragmented Chained Cluster

Deleted Files

Deleted Files

In both FAT and NTFS, when a file is deleted

- *The OS replaces the 1st character in the file name with the character* **0xE5**
- *Displayed by many forensic tools as either* **?** *or* **σ**
- *This tells the OS that the file is no longer available*

But it leaves the file intact

- *Forensic and other software tools can recover the file*

The space that the file occupies becomes unallocated

- *Being unallocated, it can be overwritten by other files*
- *In a FATfs the cluster chain is set to zero*