



**Intelligent
Computer
Solutions**

FAT File System Overview

**Techno Forensics 2007 Conference
Gaithersburg, MD**

October 30th, 2007

Understanding Data Storage

Learning to Count

- The Decimal Number 2,468 has four digits.

- Expressing Explicitly

$$(2 * 1000) + (4 * 100) + (6 * 10) + (8 * 1) =$$
$$2000 + 400 + 60 + 8 = 2468$$

- Expressing with Powers of 10

$$(2 * 10^3) + (4 * 10^2) + (6 * 10^1) + (8 * 10^0) =$$
$$2000 + 400 + 60 + 8 = 2468$$

Bits - Counting in Binary

- The word bit is a shortening of the words “Binary digI” and is generally the smallest readable unit on a computer.
- Starting at zero and going through 20, counting in decimal and binary looks like this:

0 =	0	10 =	1010
1 =	1	11 =	1011
2 =	10	12 =	1100
3 =	11	13 =	1101
4 =	100	14 =	1110
5 =	101	15 =	1111
6 =	110	16 =	10000
7 =	111	17 =	10001
8 =	1000	18 =	10010
9 =	1001	19 =	10011
		20 =	10100

- Consider one byte to be a sequence of 8 bits.
- With 8 bits in a byte, you can represent 256 values ranging from 0 to 255, as shown here:

0 = 00000000

1 = 00000001

2 = 00000010

...

254 = 11111110

255 = 11111111

- **ASCII is the abbreviation for American Standard Code for Information Interchange.**
- **ASCII is just one of a number of standardized sets of computer characters.**
- **The Extended ASCII Character Set contains 256 values.**

Standard ASCII Codes

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(72	48	H	104	68	h
9	09	Horizontal tab	41	29)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	61	3D	=	93	5D]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

Extended ASCII Codes

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	Ł	224	E0	α
129	81	ù	161	A1	í	193	C1	ł	225	E1	β
130	82	é	162	A2	ó	194	C2	ŧ	226	E2	Γ
131	83	â	163	A3	ú	195	C3	ł̇	227	E3	π
132	84	ä	164	A4	ñ	196	C4	—	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	†	229	E5	σ
134	86	ã	166	A6	ª	198	C6	‡	230	E6	μ
135	87	ç	167	A7	º	199	C7	‡	231	E7	τ
136	88	ê	168	A8	¿	200	C8	Ł	232	E8	Φ
137	89	ë	169	A9	ƒ	201	C9	Ŧ	233	E9	Θ
138	8A	è	170	AA	ſ	202	CA	Ł̇	234	EA	Ω
139	8B	ì	171	AB	½	203	CB	Ŧ	235	EB	ϛ
140	8C	í	172	AC	¾	204	CC	‡	236	EC	∞
141	8D	î	173	AD	ı	205	CD	=	237	ED	∞
142	8E	Ë	174	AE	«	206	CE	‡	238	EE	ε
143	8F	Ě	175	AF	»	207	CF	Ł̇	239	EF	Π
144	90	É	176	B0	⋯	208	DO	Ł̇	240	FO	≡
145	91	æ	177	B1	⋮	209	D1	Ŧ	241	F1	±
146	92	Æ	178	B2	⋭	210	D2	Ŧ	242	F2	≥
147	93	ó	179	B3		211	D3	Ł̇	243	F3	≤
148	94	ö	180	B4	†	212	D4	Ł̇	244	F4	[
149	95	ò	181	B5	‡	213	D5	Ŧ	245	F5]
150	96	û	182	B6	‡	214	D6	Ŧ	246	F6	÷
151	97	ù	183	B7	Ŧ	215	D7	‡	247	F7	≈
152	98	ÿ	184	B8	Ŧ	216	D8	‡	248	F8	°
153	99	Ö	185	B9	‡	217	D9	Ŧ	249	F9	•
154	9A	Û	186	BA		218	DA	Ŧ	250	FA	·
155	9B	◊	187	BB	Ŧ	219	DB	■	251	FB	√
156	9C	£	188	BC	Ł̇	220	DC	■	252	FC	π
157	9D	¥	189	BD	Ł̇	221	DD	■	253	FD	²
158	9E	ℳ	190	BE	Ŧ	222	DE	■	254	FE	■
159	9F	f	191	BF	Ŧ	223	DF	■	255	FF	□

Try this:

- **Open up a new file in Notepad and insert the sentence, "Four score and seven years ago".**
- **Save the file to disk under the name getty.txt.**
- **Use the explorer and look at the size of the file. You will find that the file has a size of 30 bytes on disk: 1 byte for each character.**
- **If you add another word to the end of the sentence and re-save it, the file size will jump to the appropriate number of bytes. Each character consumes a byte.**

Hexadecimal

- **Hexadecimal describes a base-16 number system.**
- **The hexadecimal numbers are 0-9 and then use the letters A-F.**
- **Hexadecimal is a convenient way to express binary numbers in modern computers in which a byte is almost always defined as containing eight binary digits.**

Hex Editor

The screenshot shows the WinHex application window. The menu bar includes File, Edit, Search, Position, View, Tools, Specialist, Options, File Manager, Window, and Help. The toolbar contains various icons for file operations and editing. The main window is divided into three panes:

- Left Pane:** Displays floppy disk information for 'Floppy disk 0'.
 - Default Edit Mode: original
 - State: original
 - Undo level: 0
 - Undo reverses: n/a
 - Total capacity: 1.4 MB (1,474,560 bytes)
 - Number of tracks: 80
 - Number of sides: 2
 - Sectors per track: 18
 - Track No.: 0
 - Side No.: 0
 - Sector No.: 1
 - Window #: 1
 - No. of windows: 1
 - Mode: Text
 - Character set: ANSI ASCII
 - Offsets: hexadecimal
 - Bytes per page: 36x16=576
 - Clipboard: available
 - TEMP folder: 60.0 GB free
 - UME: ~1\ADMINI~1\LOCALS~1\Temp
- Center Pane:** A hex editor view showing a table of data from 'Floppy disk 0'.

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	EB	34	90	49	42	4D	20	20	33	2E	33	00	02	01	01	00	4 IBM 3.3.....
00000010	02	E0	00	40	0B	F0	09	00	12	00	02	00	00	00	00	00	.à.@.š.....
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	01	00	FA	33	C0	8E	D0	BC	00	7C	16	07ú3Á B% ..
00000040	BB	78	00	36	C5	37	1E	56	16	53	BF	2B	7C	B9	0B	00	>>x.6Á7.V.Sç+ '..
00000050	FC	AC	26	80	3D	00	74	03	26	8A	05	AA	8A	C4	E2	F1	ü-& =t.& .# Áãñ
00000060	06	1F	89	47	02	C7	07	2B	7C	FB	CD	13	72	67	A0	10	.. G.Ç.+ úí.rg
00000070	7C	98	F7	26	16	7C	03	06	1C	7C	03	06	0E	7C	A3	3F	=& é?
00000080	7C	A3	37	7C	B8	20	00	F7	26	11	7C	8B	1E	0B	7C	03	é7 , +&
00000090	C3	48	F7	F3	01	06	37	7C	BB	00	05	A1	3F	7C	E8	9F	ÁH÷ó..7 ».. ? è
000000A0	00	B8	01	02	E8	B3	00	72	19	8B	FB	B9	0B	00	BE	D6	.. è³.r.ú¹.Ö
000000B0	7D	F3	A6	75	0D	8D	7F	20	BE	E1	7D	B9	0B	00	F3	A6	}ó u. á¹..ó
000000C0	74	18	BE	77	7D	E8	6A	00	32	E4	CD	16	5E	1F	8F	04	t.w}èj.2áí.°
000000D0	8F	44	02	CD	19	BE	C0	7D	EB	EB	A1	1C	05	33	D2	F7	D.í.Á}èèi..30÷
000000E0	36	0B	7C	FE	C0	A2	3C	7C	A1	37	7C	A3	3D	7C	BB	00	6. pÁ<< 7 é= »..
000000F0	07	A1	37	7C	E8	49	00	A1	18	7C	2A	06	3B	7C	40	38	.. 7 èI.i. *.; @8
00000100	06	3C	7C	73	03	A0	3C	7C	50	E8	4E	00	58	72	C6	28	< s.< PèN.XrÆ(
00000110	06	3C	7C	74	0C	01	06	37	7C	F7	26	0B	7C	03	D8	EB	< t...7 ÷& . .0è
00000120	D0	8A	2E	15	7C	8A	16	FD	7D	8B	1E	3D	7C	EA	00	00	D ... í.ý) .= è..
00000130	70	00	AC	0A	C0	74	22	B4	0E	BB	07	00	CD	10	EB	F2	p..Át"'.>.. I.èò
00000140	33	D2	F7	36	18	7C	FE	C0	88	16	3B	7C	33	D2	F7	36	30÷6. pÁ ; 30÷6
00000150	1A	7C	88	16	2A	7C	A3	39	7C	C3	B4	02	8B	16	39	7C	.. .* é9 Á'. .9
00000160	B1	06	D2	E6	0A	36	3B	7C	8B	CA	86	E9	8A	16	FD	7D	±.0æ.6; IÉ é .ý)
00000170	8A	36	2A	7C	CD	13	C3	0D	0A	4E	6F	6E	2D	53	79	73	I6* í.Á..Non-Sys
00000180	74	65	6D	20	64	69	73	6B	20	6F	72	20	64	69	73	6B	tem disk or disk
00000190	20	65	72	72	6F	72	0D	0A	52	65	70	6C	61	63	65	20	error..Replace
000001A0	61	6E	64	20	73	74	72	69	6B	65	20	61	6E	79	20	6B	and strike any k
000001B0	65	79	20	77	68	65	6E	20	72	65	61	64	79	0D	0A	00	ey when ready...
000001C0	0D	0A	44	69	73	6B	20	42	6F	6F	74	20	66	61	69	6C	..Disk Boot fail
000001D0	75	72	65	0D	0A	00	49	42	4D	42	49	4F	20	20	43	4F	ure...IBMBIO CO
000001E0	4D	49	42	4D	44	4F	53	20	20	43	4F	4D	00	00	00	00	MIBMDOS COM...
000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AAUª
00000200	F0	FF	FF	00	00	00	00	00	00	00	00	00	00	00	00	00	šÿÿ.....
00000210	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000220	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000230	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
- Right Pane:** A data interpreter window showing the ASCII representation of the hex data. It contains text such as "IBM 3.3", "à.@.š", "ú3Á|B%", "x.6Á7.V.Sç+", "ü-&|=t.&|.#|Áãñ", "G.Ç.+|úí.rg", "|=&|.|...|...|é?", "|é7|,|+&|.||...|", "ÁH÷ó..7|»..|?|è|", "è³.r.ú¹.Ö", "ó|u.|| á¹..ó|", "t.w}èj.2áí.°||", "|D.í.Á}èèi..30÷", "6.|pÁ<<|7|é=|»..", "7|èI.i.|*.;|@8", "<|s.<|PèN.XrÆ(", "<|t...7|÷&|.|.0è", "D|...|í.ý)|.=|è..", "p..Át"'.>..|I.èò", "30÷6.|pÁ||;|30÷6", ".*|é9|Á'.|.9|", "±.0æ.6;|IÉ|é|.ý)", "I6*|í.Á..Non-Sys", "tem disk or disk", "error..Replace", "and strike any k", "ey when ready...", "..Disk Boot fail", "ure...IBMBIO CO", "MIBMDOS COM...", ".....Uª", "šÿÿ.....", ".....", ".....", ".....".

Hex Editor Warning

- **When editing files of a certain type (for instance executable files), it is essential not to change the file size.**
- **Moving the addresses of executable code and included data results in severely damaging such files.**
- **It is quite safe to edit text passages in a file.**
- **It is recommendable to create backup files before editing.**

Wiping and Initializing

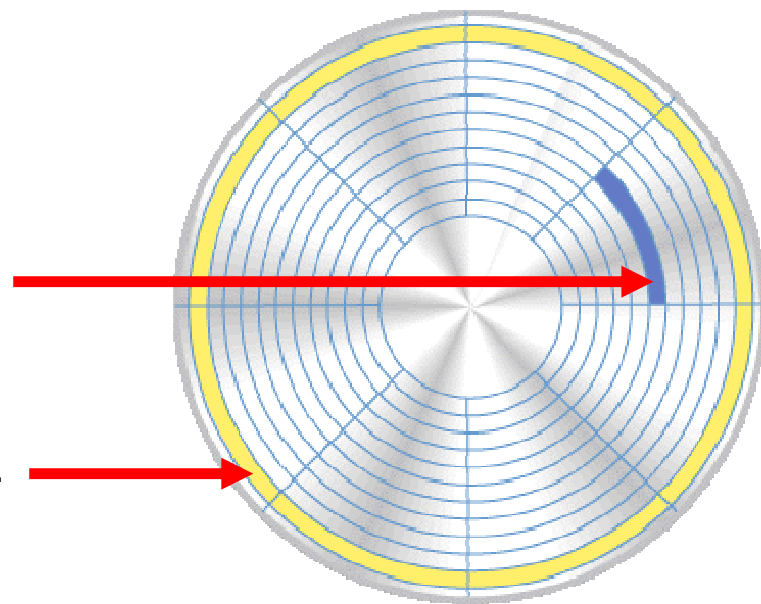
- **Use WinHex to Wipe a floppy**
- **Tools → Open Disk → Logical Drive, Removable medium (A:)**
- **Edit Menu → Fill Disk Sectors**
- **Fill with hex values = 00, Passes: Pass #1**
- **Changes have only been written to buffer until you save changes, File → Save Sectors**
- **Verify Wipe by closing WinHex, reopen WinHex, Tools → Open Disk → Physical Media, Floppy disk 0**

Sectors and Tracks

- Data is stored on the surface of a platter in sectors and tracks.
- Tracks are concentric circles, and sectors are pie-shaped wedges on a track

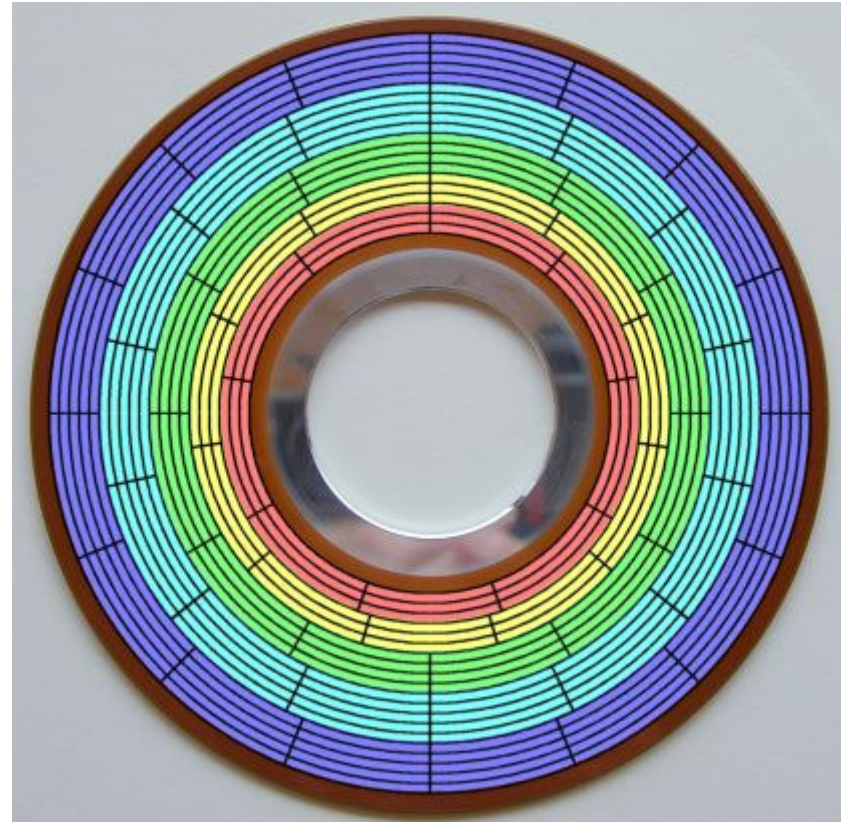
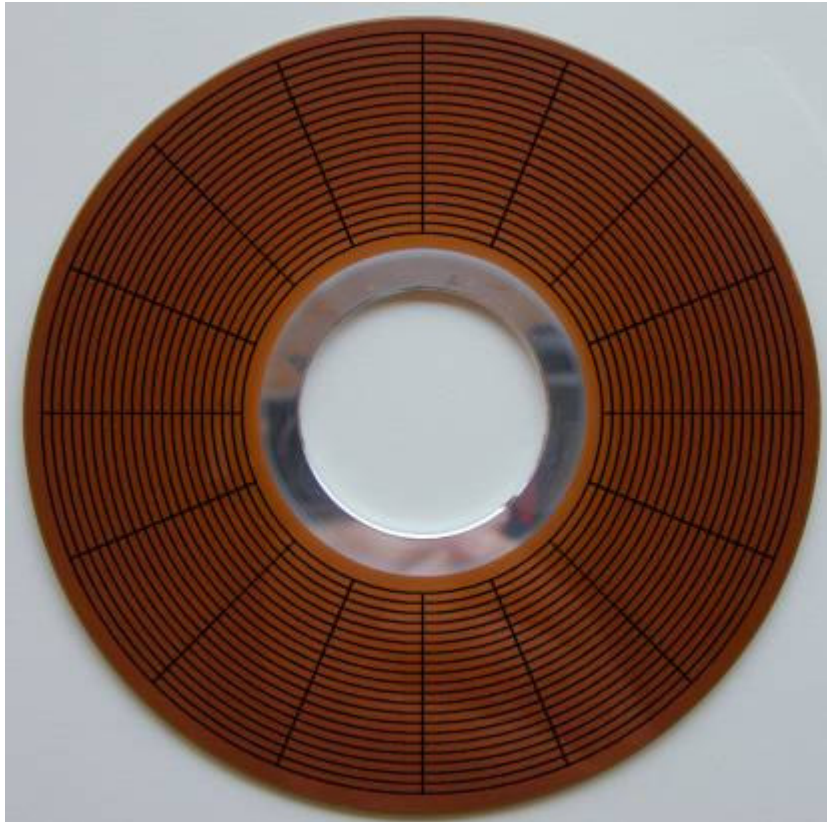
A typical sector is shown in blue.

A typical track is shown in yellow.



©2000 How Stuff Works

Zoned Bit Recording



The PC Guide (<http://www.PCGuide.com>)

Low-Level Formatting

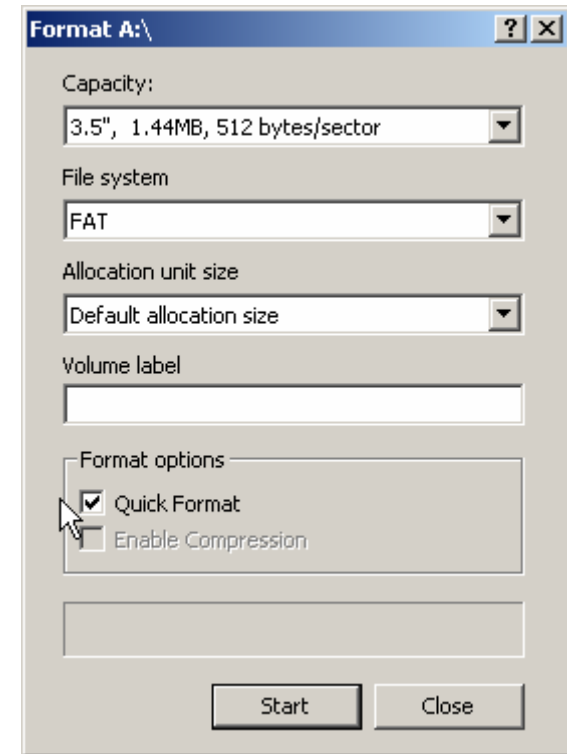
- The process of low-level formatting a drive establishes the tracks and sectors on the platter.
- The starting and ending points of each sector are written onto the platter.
- This process prepares the drive to hold blocks of bytes.

High-Level Formatting

- **High-level formatting then writes the file-storage structures, like the file-allocation table, into the sectors.**
- **This process prepares the drive to hold files.**

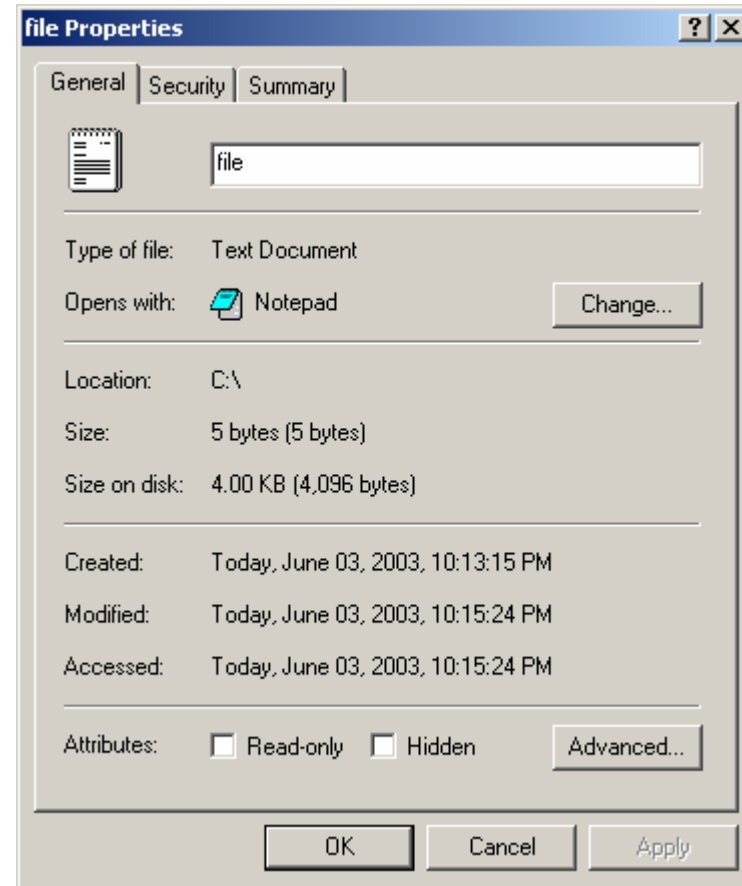
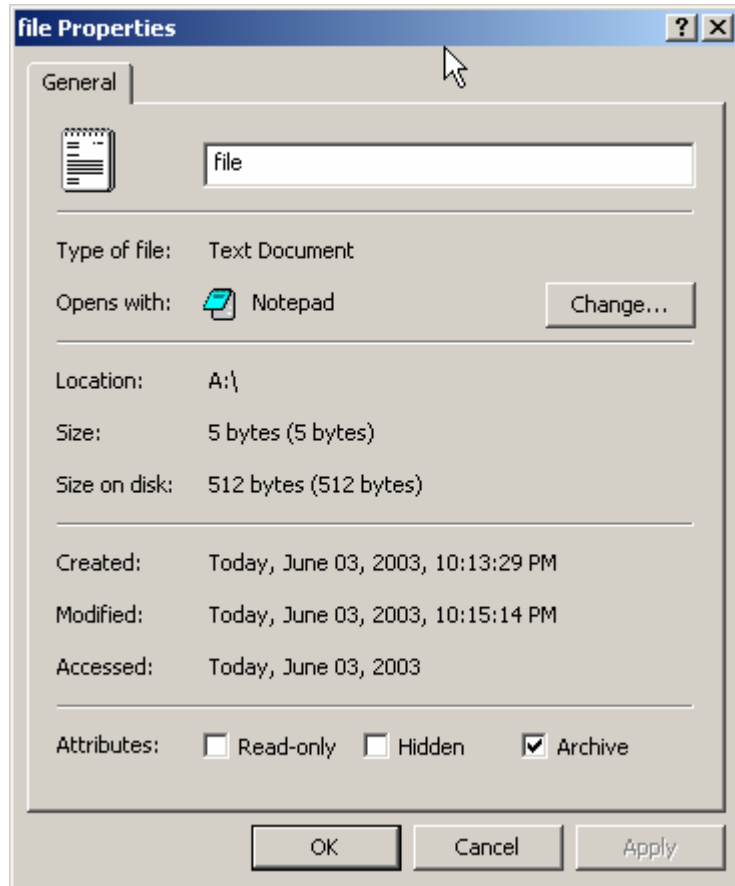
Floppy Format Process

- **Floppy diskettes** use the **FAT12** file system, the low level-format and high-level format take place at the same time and they are performed by the operating system.
- If you check the box “Quick Format” under “Format Options” you are only performing a **high-level format**, which is why it’s quicker, essentially erasing only the File Allocation Table (FAT).



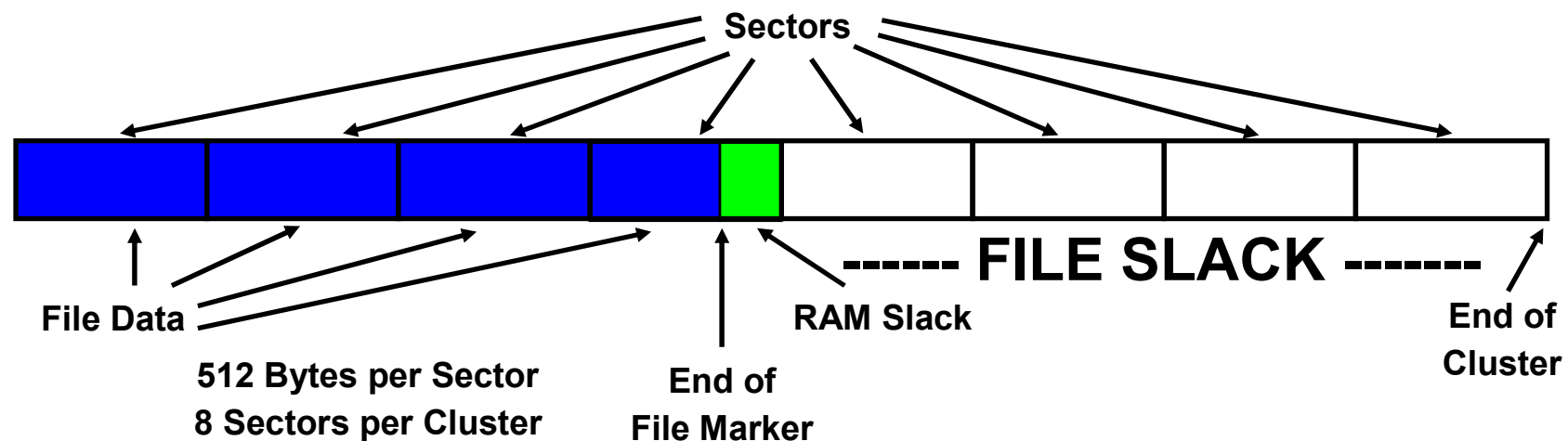
- **All Microsoft operating systems read and write in blocks of data called clusters.**
- **A cluster is simply a group of sectors.**
- **Clusters are defined during the high level format which is performed by the operating system.**

Sectors per Cluster



File Slack

- File slack is the data that resides from the end of the file to the end of a cluster.
- File slack can be from 1byte to the size of a cluster minus one byte.



Big Endian / Little Endian

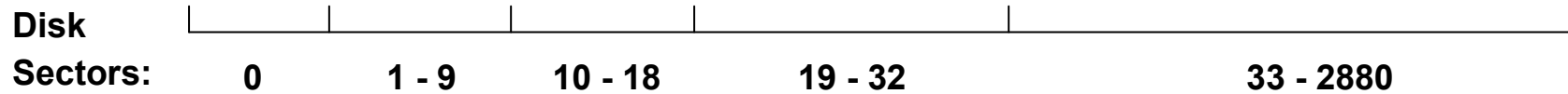
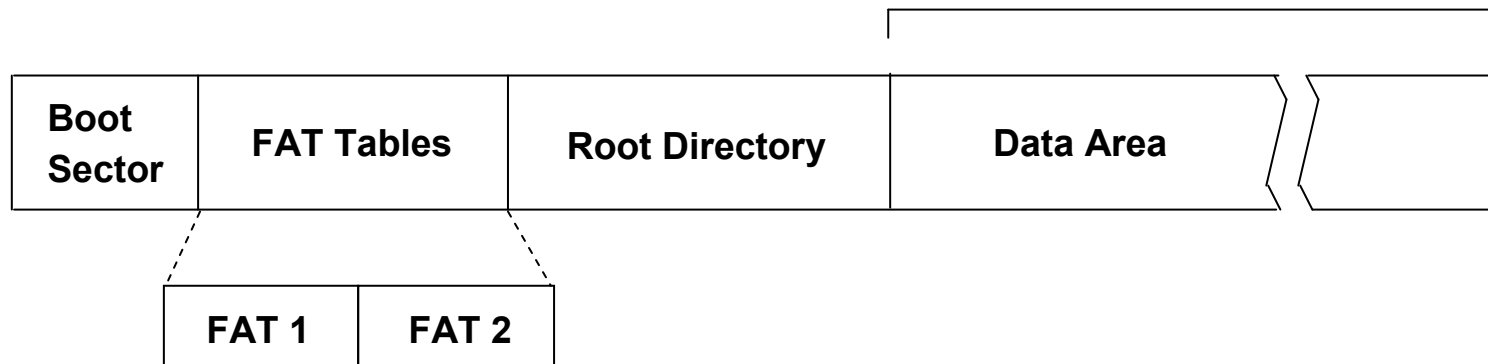
- **Big Endian and Little Endian are terms that describe the order in which a sequence of bytes are stored in computer memory.**

Hex 12 34 56 78

Big Endian	Little Endian
Byte 0: 12	Byte 0: 78
Byte 1: 34	Byte 1: 56
Byte 2: 56	Byte 2: 34
Byte 3: 78	Byte 4: 12

DOS Disk Layout

File Clusters: 2 - 2849



Boot Record Diagram

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
EB	3E	90	4D	53	57	49	4E	34	2E	31	00	02	04	01	00	è>MSWIN4.1...
02	00	02	00	00	F8	00	01	3F	00	F0	00	60	75	0C	00	...z...?è.u...
10	EC	03	00	80	00	29	1E	3C	D9	19	4E	4F	20	4E	41	.i...e...<Û.NO NA
4D	45	20	20	20	20	46	41	54	31	36	20	20	20	F1	7D	ME FAT16
FA	33	C9	8E	D1	BC	FC	7B	16	07	BD	78	00	C5	76	00	ú3É+Ñmu{...x.Áv.
1E	56	16	55	BF	22	05	89	7E	00	89	4E	02	B1	0B	FC	.V.Uz".z~.zN.±ú
F3	A4	06	1F	BD	00	7C	C6	45	FE	0F	8B	46	18	88	45	óP... ÆEþ.<F.°E
F9	FB	38	66	24	7C	04	CD	13	72	3C	8A	46	10	98	F7	úú8f\$.Í.r<ŠF."+
66	16	03	46	1C	13	56	1E	03	46	0E	13	D1	50	52	89	f...F...V...F...ÑPRz
46	FC	89	56	FE	B8	20	00	8B	76	11	F7	E6	8B	5E	0B	FuZVp...<v.+æ<^
03	C3	48	F7	F3	01	46	FC	11	4E	FE	5A	58	BB	00	07	.ÅH+ó.Fu.NpZK>>.
8B	FB	B1	01	E8	94	00	72	47	38	2D	74	19	B1	0B	56	<ú±.è".rG8-t.±.V
8B	76	3E	F3	A6	5E	74	4A	4E	74	0B	03	F9	83	C7	15	<v>ó ^tJNt...úfÇ.
3B	FB	72	E5	EB	D7	2B	C9	B8	D8	7D	87	46	3E	3C	D8	:úráèx+E,0}+F><@
75	99	BE	80	7D	AC	98	03	F0	AC	84	C0	74	17	3C	FF	u™%€)-.è-.Àt.<ý
74	09	B4	0E	BB	07	00	CD	10	EB	EE	BE	83	7D	EB	E5	t...>>...Í.èi%f}èä
BE	81	7D	EB	E0	33	C0	CD	16	5E	1F	8F	04	8F	44	02	z+}èä3Áí.^.+.D.
CD	19	BE	82	7D	8B	7D	0F	83	FF	02	72	C8	8B	C7	48	Í.%,}<}fÿ.rÈ<ÇH
48	8A	4E	0D	F7	E1	03	46	FC	13	56	FE	BB	00	07	53	HŠN.+á.Fü.Vp>>..S
B1	04	E8	16	00	5B	72	C8	81	3F	4D	5A	75	A7	81	BF	±.è...[rÈ+?MZu\$+è
00	02	42	4A	75	9F	EA	00	02	70	00	50	52	51	91	92	...BJuÿè...p.PRQ"
33	D2	F7	76	18	91	F7	76	18	42	87	CA	F7	76	1A	8A	30+v...+v.B+È+v.Š
F2	8A	56	24	8A	E8	D0	CC	D0	CC	0A	CC	B8	01	02	CD	óŠV\$SèDìDì.ì...í
13	59	5A	58	72	09	40	75	01	42	03	5E	0B	E2	CC	C3	.YZXr.@u.B.^áíÄ
03	18	01	27	0D	0A	49	6E	76	61	6C	69	64	20	73	79	...Invalid sy
73	74	65	6D	20	64	69	73	6B	FF	0D	0A	44	69	73	6B	stem diskÿ...Disk
20	49	2F	4F	20	65	72	72	6F	72	FF	0D	0A	52	65	70	I/O errorÿ...Rep
6C	61	63	65	20	74	68	65	20	64	69	73	6B	2C	20	61	lace the disk, a
6E	64	20	74	68	65	6E	20	70	72	65	73	73	20	61	6E	nd then press an
79	20	6B	65	79	0D	0A	00	49	4F	20	20	20	20	20	20	y key...IO
53	59	53	4D	53	44	4F	53	20	20	20	53	59	53	80	01	SYSMSDOS SYSÈ.
00	57	49	4E	42	4F	4F	54	20	53	59	53	00	00	55	AA	.WINBOOT SYS...Ûª

- Jumpcode
- OEM / ID Name
- Bytes per Sector
- Sectors per Allocation
- Reserved Sectors
- Number of FATs
- Root Entries
- Total Sectors
- Media Type
- Sectors per FAT
- Sectors per Track
- Number of Heads
- Hidden Sectors
- Total Sectors
- Drive ID
- NT Reserved
- Extended Boot Signature
- Volume Serial Number
- Volume / Partition Number
- Fat Type
- Executable Boot (strap) Code
- Executable Signature

Boot Record Diagram (cont.)

- Jumpcode (= 3 bytes) - the offset jump to the boot(strap) executable code plus a nop. from the disk: eb,3e,90 -> translates to: |jumpshort(to)|offset 3e|no operation|
- OEM / ID Name (= 8 bytes)- some indication of what system formatted the partition, not checked, but set for compatibility; mswin4.0, mswin4.1, and msdos5.0 as formatted by windows 95, 98, and XP respectfully.
- Bytes per Sector (= 2 bytes) - normally set to 512 bytes; from the disk: 00,02 flip -> 02,00 convert to decimal = 512 bytes.
- Sectors per Cluster (= 1 byte) - states the number of sectors per cluster.
- Reserved Sectors (= 2 bytes) - states the number of reserved sectors, on fat12/16: 01,00 flip -> 00,01 convert to decimal = 1 sector – this is the boot record.
- Number of FATs (= 1 byte) - states the number fats used, normally set to 2 in case of bad sectors, which could lead to data errors, however >=1 is also valid.
- Root Entries (= 2 bytes) - states the maximum number of 32 byte entries in the root directory; unused for fat32 and set to 00,00; however for fat16: 00,02 flip -> 02,00 convert to decimal = 512; $512 * 32(\text{bytes}) = 16384$ bytes - data is stored after this point.

Volume Serial Number

- **When a partition is formatted, it will display the newly assigned serial such as: 15e7-2a35.**
- **The Volume Serial Number is calculated by combining the date and time at the point of format, it is an unique identifier to keep track of drives in use.**
- **It is not possible to retrieve the date and time from the serial number.**

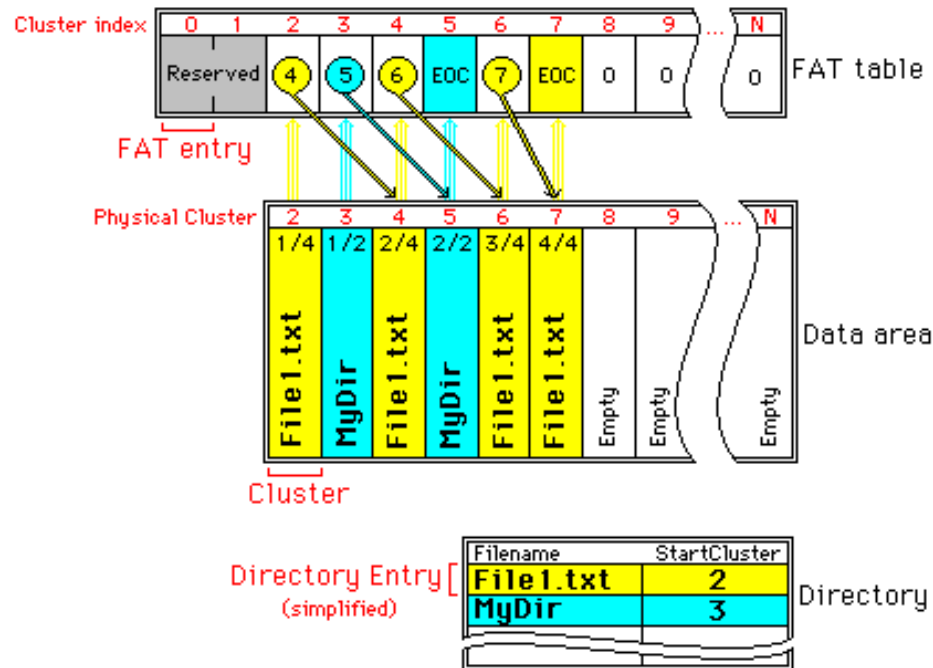
Volume Serial Number (cont.)

- The first byte is calculated as follows:
milliseconds + days -> convert to hex
 $50 + 3 = 53 \rightarrow = 35$

The second byte is calculated as follows:
months + seconds -> convert to hex
 $10 + 32 = 42 \rightarrow = 2a$

The last two bytes are calculated as follows:
(hours [if pm + 12] * 256) + minutes + years -> convert to
hex & flip
 $(2 + 12 = 14 * 256 = 3584) + 22 + 2001 = 5607 \rightarrow 15,e7 \rightarrow$
 $e7,15$

Organization of the FAT



- A 12-bit FAT entry may have the following values:
 - 0x000 Unused cluster.
 - 0xFF0-0xFF6 Reserved cluster (like FAT entries 0 and 1).
 - 0xFF7 Bad Cluster (contains errors and should not be used).
 - 0xFF8-0xFFFF End of file/directory, also called EOC (end of cluster chain).
 - other numbers are pointers (indexes) to the next cluster in the file/directory.

File Allocation Table

- **The File Allocation Table is used to track which clusters have been allocated to a specific file**
- **The FAT is relied upon by the operating system much like a card catalog system is used in a library to locate a book**
- **References in the FAT act as pointers and they point to clusters by numeric reference**

Understand the FAT Table

● AB CD EF → DAB EFC

How Files are Stored

- **When a file is created three things occur:**
 - 1. An entry is made into the File Allocation Table to indicate where the actual data is stored in the Data Area.**
 - 2. A Directory Entry is made to indicate file name, size, the link to the FAT, and other information.**
 - 3. The data is written to the Data Area.**

How Files are Deleted

- **When a file is deleted only two things occur:**
 - 1. The File Allocation Table entry for that particular file is zeroed out and shown as available for use by a new file.**
 - 2. The first character of the Directory Entry file is changed to a special character (E5 HEX).**
 - 3. Nothing is done to the Data Area.**

How Files are Recovered

- **Recovery of a FAT file system file can be accomplished in 2 ways:**
 - 1. The File Allocation Table entry for that particular file is linked to the particular location in the data area where the file data is stored.**
 - 2. The first character of the Directory Entry file is changed to a legal character.**
 - 3. Nothing is done to the Data Area.**

How Files are Recovered (cont.)

- **Recovery of a FAT file system file can be accomplished in 2 ways (cont.):**
 - 1. The data can be “carved” from unallocated space using specialized utilities.**