FAT32 Boot Sector, Locating Files and Dirs

Classes COP4610 / CGS5765 Florida State University

Outline

Recap of last week's lecture

- Introduction to project 3
- Introduction to FAT32 structure
- Starting Project 3
 - How to parse the boot sector
 - Finding the root directory and files

Project 3

Reminder: It's a group project

- 3 people in each group, everyone gets the same grade
- Email your group member's name before the next Friday
- Also email if you are looking for a group

Recap – Intro to Project 3 and FAT32

Project 3

 You will create a user-space utility to manipulate a FAT32 file system image
 No more kernel programming!

FAT32 Manipulation Utility

Utility only recognizes the following built-in commands:

open
close
create
rm
rimdir
size
read
write

Terminology

- Byte 8 bits of data, the smallest addressable unit in modern processors
- Sector Smallest addressable unit on a storage device. Usually this is 512 bytes
- Cluster FAT32-specific term. A group of sectors representing a chunk of data
- FAT Stands for *file allocation table* and is a map of files to data

FAT32 Disk Layout

3 main regions...





Reserved Region

Reserved Region – Includes the boot sector, the extended boot sector, the file system information sector, and a few other reserved sectors



FAT Region

 FAT Region – A map used to traverse the data region. Contains mappings from cluster locations to cluster locations



Data Region

 Data Region – Using the addresses from the FAT region, contains actual file/directory data



FAT32 Disk Layout

/ Volu / Rese	ime ID erved Sect	ors	
	FAT #1	EAT #2	
		- TAT#2	
		Clusters (Files and	Directories)
			,
			Small Unused Space

In this project's context, Volume ID basically means the Boot Sector

Where to begin?

- Mount the file system image with the OS FAT32 driver and take a look around
- Find the FAT32 spec from Microsoft in the lab website, have a look in it
 - This document is written for those who already know the FAT32 structure well, so may seem a bit difficult to understand at first.
 - However, it will be very useful once you start coding

Hint

- As you work, it might make sense to first take a look at the raw file system image
- Hexedit to the rescue!

Hexedit

\$> hexedit [filename]

- View files in hexadecimal or ASCII
- Why wouldn't you want to view the file system image file in your regular editor?

Hexedit

🛃 user@cop4610: ~																	
00000000	EB	58	90	6D	6B	64	6F	73	66	73	00	00	02	01	20	00	.X.mkdosfs ^
00000010	02	00	00	00	00	F8	00	00	20	00	40	00	00	00	00	00	
00000020	00	00	02	00	F1	03	00	00	00	00	00	00	02	00	00	00	
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000040	00	00	29	6E	FA	2E	43	20	20	20	20	20	20	20	20	20)nC
00000050	20	20	46	41	54	33	32	20	20	20	0E	1F	BE	77	7C	AC	FAT32w .
00000060	22	C0	74	0B	56	B4	0E	BB	07	00	CD	10	5E	EB	FO	32	".t.V^2
00000070	E4	CD	16	CD	19	EB	FE	54	68	69	73	20	69	73	20	6E	This is n
00000080	6F	74	20	61	20	62	6F	6F	74	61	62	6C	65	20	64	69	ot a bootable di
00000090	73	6B	2E	20	20	50	6C	65	61	73	65	20	69	6E	73	65	sk. Please inse
0A000000	72	74	20	61	20	62	6F	6F	74	61	62	6C	65	20	66	6C	rt a bootable fl
000000B0	6F	70	70	79	20	61	6E	64	0D	0A	70	72	65	73	73	20	oppy andpress
000000000	61	6E	79	20	6B	65	79	20	74	6F	20	74	72	79	20	61	any key to try a
000000D0	67	61	69	6E	20	2E	2E	2E	20	0D	0A	00	00	00	00	00	gain
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	

He	X¢	ec	li	t		L uml r	ine ber nex	s in									
B user@cop4610:	~			_													
0000000	EB	58	90	6D	6B	64	6F	73	66	73	00	00	02	01	20	00	.X.mkdosfs ^
00000010	02	00	00	00	00	F8	00	00	20	00	40	00	00	00	00	00	· · · · · · · · · · · · · · · · · · ·
00000020	00	00	02	00	F1	03	00	00	00	00	00	00	02	00	00	00	
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000040	00	00	29	6E	FA	2E	43	20	20	20	20	20	20	20	20	20)nC
00000050	20	20	46	41	54	33	32	20	20	20	0E	1F	BE	77	7C	AC	FAT32w .
00000060	22	C0	74	0B	56	B4	0E	BB	07	00	CD	10	5E	EB	FO	32	".t.V^2
00000070	E4	CD	16	CD	19	EB	FE	54	68	69	73	20	69	73	20	6E	This is n
00000080	6F	74	20	61	20	62	6F	6F	74	61	62	6C	65	20	64	69	ot a bootable di
00000090	73	6B	2E	20	20	50	6C	65	61	73	65	20	69	6E	73	65	sk. Please inse
04000000	72	74	20	61	20	62	6F	6F	74	61	62	6C	65	20	66	6C	rt a bootable fl
000000В0	6F	70	70	79	20	61	6E	64	0D	0A	70	72	65	73	73	20	oppy and press
000000000	61	6E	79	20	6B	65	79	20	74	6F	20	74	72	79	20	61	any key to try a
000000000	67	61	69	6E	20	2E	2E	2E	20	OD	AO	00	0.0	00	00	00	gain
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	
00000170	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	
00000180	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	

He	X¢	ec	li 1	t					Со	nte hex	nt ii <	า					
🛃 user@cop4610: -	~												-				
00000000	EB	58	90	6D	6B	64	6F	73	66	73	00	00	02	01	20	00	.X.mkdosfs ^
00000010	02	00	00	00	00	F8	00	00	20	00	40	00	00	00	00	00	
00000020	00	00	02	00	F1	03	00	00	00	00	00	00	02	00	00	00	
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000040	00	00	29	6E	FA	2E	43	20	20	20	20	20	20	20	20	20)nC
00000050	20	20	46	41	54	33	32	20	20	20	0E	1F	BE	77	7C	AC	FAT32w .
00000060	22	C0	74	0B	56	B4	0E	BB	07	00	CD	10	5E	EB	FO	32	".t.V^2
00000070	E4	CD	16	CD	19	EB	FE	54	68	69	73	20	69	73	20	6E	This is n
00000080	6F	74	20	61	20	62	6F	6F	74	61	62	6C	65	20	64	69	ot a bootable di
00000090	73	6B	2E	20	20	50	6C	65	61	73	65	20	69	6E	73	65	sk. Please inse
000000A0	72	74	20	61	20	62	6F	6F	74	61	62	6C	65	20	66	6C	rt a bootable fl
000000B0	6F	70	70	79	20	61	6E	64	0D	0A	70	72	65	73	73	20	oppy andpress
000000000	61	6E	79	20	6B	65	79	20	74	6F	20	74	72	79	20	61	any key to try a
000000D0	67	61	69	6E	20	2E	2E	2E	20	0D	0A	00	00	00	00	00	gain
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	

He	XC	ec	li	t													Content in printable ASCII
🛃 user@cop4610: ~	1																
00000000	EB	58	90	6D	6B	64	6F	73	66	73	00	00	02	01	20	00	.X.mkdosfs ^
00000010	02	00	00	00	00	F8	00	00	20	00	40	00	00	00	00	00	
00000020	00	00	02	00	F1	03	00	00	00	00	00	00	02	00	00	00	
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000040	00	00	29	6E	FA	2E	43	20	20	20	20	20	20	20	20	20)nC
00000050	20	20	46	41	54	33	32	20	20	20	0E	1F	BE	77	7C	AC	FAT32w .
00000060	22	C0	74	0B	56	B4	0E	BB	07	00	CD	10	5E	EB	FO	32	".t.V^2
00000070	E4	CD	16	CD	19	EB	FE	54	68	69	73	20	69	73	20	6E	This is n
08000000	6F	74	20	61	20	62	6F	6F	74	61	62	6C	65	20	64	69	ot a bootable di
00000090	73	6B	2E	20	20	50	6C	65	61	73	65	20	69	6E	73	65	sk. Please inse
0A000000	72	74	20	61	20	62	6F	6F	74	61	62	6C	65	20	66	6C	rt a bootable fl
000000B0	6F	70	70	79	20	61	6E	64	0D	A0	70	72	65	73	73	20	oppy andpress
000000000	61	6E	79	20	6B	65	79	20	74	6F	20	74	72	79	20	61	any key to try a
000000D0	67	61	69	6E	20	2E	2E	2E	20	0D	0A	00	00	00	00	00	gain
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	

Hexadecimal Hints

- Hex is base 16 one hexadecimal can represent 0-15
- It takes 4 binary bits to represent values 0-15
 - **0** 0000 = 0
 - □ 1111 = 15

Hexadecimal Hints

- If it takes 4 bits to represent one hexadecimal number, it takes 8 bits to represent two hexadecimal numbers
 - 8 bits = 1 byte
- Two hex numbers together symbolize one byte
 - That's why hex numbers are in groups of two

Endianness

- FAT32 is represented in little endian byte order
 - Reading left to right, you encounter leastsignificant byte first
 - What 32-bit number is this? 0x0000040 or 0x40000000?

💕 user@cop4610: ~							-					-					
00000000	<mark>E</mark> B	58	90	6D	6B	64	6F	73	66	73	00	00	02	01	20	00	.X.mkdosfs ^
00000010	02	00	00	00	00	F8	00	00	20	00	40	00	00	00	00	00	
00000020	00	00	02	00	F1	03	00	00	00	00	00	00	02	00	00	00	
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000040	00	00	29	6E	FA	2E	43	20	20	20	20	20	20	20	20	20)nC
00000050	20	20	46	41	54	33	32	20	20	20	0E	1F	BE	77	7C	AC	FAT32w .
00000060	22	C0	74	0B	56	В4	0E	BB	07	00	CD	10	5E	EB	F0	32	".t.V^2
00000070	E4	CD	16	CD	19	EB	FE	54	68	69	73	20	69	73	20	6E	This is n

Endianness

Why are characters in order (readable) if some numbers are not?

🛃 user@cop4610: ~													-				
00000000	EB	58	90	6D	6B	64	6F	73	66	73	00	00	02	01	20	00	.X.mkdosfs ^
00000010	02	00	00	00	00	F8	00	00	20	00	40	00	00	00	00	00	
00000020	00	00	02	00	F1	03	00	00	00	00	00	00	02	00	00	00	
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000040	00	00	29	6E	FA	2E	43	20	20	20	20	20	20	20	20	20)nC
00000050	20	20	46	41	54	33	32	20	20	20	0E	1F	BE	77	7C	AC	FAT32w .
00000060	22	C0	74	0B	56	B4	0E	BB	07	00	CD	10	5E	EB	F0	32	".t.V^2
00000070	E4	CD	16	CD	19	EB	FE	54	68	69	73	20	69	73	20	6E	This is n
08000000	6F	74	20	61	20	62	6F	6F	74	61	62	6C	65	20	64	69	ot a bootable di
00000090	73	6B	2E	20	20	50	6C	65	61	73	65	20	69	6E	73	65	sk. Please inse
000000A0	72	74	20	61	20	62	6F	6F	74	61	62	6C	65	20	66	6C	rt a bootable fl
000000B0	6F	70	70	79	20	61	6E	64	0D	0A	70	72	65	73	73	20	oppy andpress

Endianness

- You must account for little endianness across bytes when reading in numbers of size larger than one byte
 - Characters are only one byte, no re-ordering necessary

Starting Project 3

File Allocation Table (FAT)

 Contains a chain of all the clusters belonging to a particular file

 Basically a big array of **32 bit** integers (Hence the file system is called FAT32)

File Allocation Table (FAT)

- Each integer's position in the array corresponds to a cluster number
- The value stored there indicates the next cluster of the file
- An EoC value indicates the end of the cluster chain for that file

File Allocation Table (FAT)

XXXXXXX	XXXXXXXX	0000009	00000004
00000005	0000007	00000000	80000008
FFFFFFF	A0000000	000000B	00000011
000000D	000000E	FFFFFFF	00000010
00000012	FFFFFFF	00000013	00000014
00000015	00000016	FFFFFFF	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000

Roo	t D)ire	cto)ry:	
2,	9,	Α,	Β,	11	

ile	#1	•
-----	----	---

3, 4, 5, 7, 8

File #2: C, D, E

File #3:

F, 10, 12, 13, 14, 15, 16

Steps to read from a FAT32 image

- Locate, read, and extract important info from the Boot Sector
- Locate the Root Directory, get the list of files and folders
- Access the files and directories using information from the Root Directory and the FAT32 table

Parse the Boot Sector

Where to find the Boot Sector?

- First 512 bytes of the disk (or, in our case, the 'image')

Important Boot Sector Information

- Size of each region
 - BPB_BytesPerSec
 - BPB_SecPerClus
 - BPB_RsvdSecCnt
 - BPB_NumFATS
 - BPB_FATSz32
- Root directory (first directory in tree)
 - BPB_RootClus

Important Boot Sector Information

Warning: this list is not exhaustive!

- Check the "Boot Sector and BPB Structure" in MS FAT32 File System Spec for:
 - The complete list of attributes
 - Their significance
 - Where they are located within the Boot Sector

Important Boot Sector Information

Example: extracting BPB_BytesPerSector
 Offset 11, size 2 bytes
 0x0200 = 512

🛃 user@cop4610: ~	-		_													
00000000	EB 5	58 9	0 6D	6B	64	6F	73	66	73	00	00	02	01	20	00	.X.mkdosfs ^
00000010	02 0	0 00	0 00	00	F8	00	00	20	00	40	00	00	00	00	00	
00000020	00 0	0 00	2 00	F1	03	00	00	00	00	00	00	02	00	00	00	
00000030	01 0	0 00	6 00	00	00	00	00	00	00	00	00	00	00	00	00	
00000040	00 0	0 2	9 6E	FA	2E	43	20	20	20	20	20	20	20	20	20)nC
00000050	20 2	20 4	6 41	54	33	32	20	20	20	0E	1F	BE	77	7C	AC	FAT32w .
00000060	22 C	CO 7	4 OB	56	B4	0E	BB	07	00	CD	10	5E	EB	F0	32	".t.V^2
00000070	E4 C	CD 1	6 CD	19	EB	FE	54	68	69	73	20	69	73	20	6E	This is n

Next Steps

After you have parsed the boot sector and saved key values, you may want to find the root directory

Figure out the root directory cluster number from the boot sector

BPB_RootClus Offset 44, size 4 bytes 0x0000002 = 2

💕 user@cop4610: ~											- 7		-				
00000000	<mark>E</mark> B	58	90	6D	6B	64	6F	73	66	73	00	00	02	01	20	00	.X.mkdosfs ^
00000010	02	00	00	00	00	F8	00	00	20	00	40	00	00	00	00	00	
00000020	00	00	02	00	F1	03	00	00	00	00	00	00	02	00	00	00	
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000040	00	00	29	6E	FA	2E	43	20	20	20	20	20	20	20	20	20)nC
00000050	20	20	46	41	54	33	32	20	20	20	0E	1F	BE	77	7C	AC	FAT32w .
00000060	22	C0	74	0B	56	В4	0E	BB	07	00	CD	10	5E	EB	F0	32	".t.V^2
00000070	E4	CD	16	CD	19	EB	FE	54	68	69	73	20	69	73	20	6E	This is n

 Figure out where the Data Region starts in the disk

```
FirstDataSector = BPB_ResvdSectCnt + (BPB_NumFATs *
        FATsz) + RootDirSectors
Here,
FATsz = BPB_FATSz32
RootDirSectors = ((BPB_RootEntCnt * 32) +
        (BPB_BytsPerSec -1)) / BPB_BytsPerSec;
        // Becomes 0 for FAT32
```

 Figure out where the Root Directory starts in the data region, where N=cluster number

For Root Directory, N = BPB RootClus (usually 2)

Figure out where the Root Directory starts in the data region, where N=cluster number

FirstSectorofCluster = ((N - 2) * BPB_SecPerClus) +
 FirstDataSector;

- This gives the sector number of the first sector of any cluster N
- Check page 13 in MS FAT32 File System Spec for details

Read in the root directory structure located at the first sector of the root directory cluster

610:	~		-			-		-								
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
41	63	00	6F	00	64	00	65	00	00	00	OF	00	FE	FF	FF	Ac.o.d.e
FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	00	00	FF	FF	FF	FF	
43	4F	44	45	20	20	20	20	20	20	20	10	00	64	B2	6C	CODEd.1
5C	ЗD	5C	ЗD	00	00	B2	6C	5C	ЗD	03	00	00	00	00	00	\=\=1\=
41	64	00	69	00	72	00	73	00	00	00	OF	00	5D	FF	FF	Ad.i.r.s]
FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	00	00	FF	FF	FF	FF	
44	49	52	53	20	20	20	20	20	20	20	10	00	64	B2	6C	DIRSd.1
5C	ЗD	5C	ЗD	00	00	B2	6C	5C	ЗD	07	00	00	00	00	00	\=\=1\=
41	66	00	61	00	74	00	67	00	65	00	OF	00	16	6E	00	Af.a.t.g.en.
31	00	30	00	33	00	2E	00	70	00	00	00	64	00	66	00	1.0.3pd.f.
46	41	54	47	45	4E	7E	31	50	44	46	20	00	64	B2	6C	FATGEN~1PDF .d.1
5C	ЗD	5C	ЗD	00	00	B2	6C	5C	ЗD	11	00	FD	89	07	00	\=\=1\=
41	46	00	41	00	54	00	69	00	6E	00	OF	00	D2	66	00	AF.A.T.i.nf.
6F	00	2E	00	74	00	78	00	74	00	00	00	00	00	FF	FF	ot.x.t
46	41	54	49	4E	46	4F	20	54	58	54	20	00	64	B2	6C	FATINFO TXT .d.1
5C	3D	5C	3D	00	00	B2	6C	5C	ЗD	D6	03	35	01	00	00	\=\=1\=5
41	66	00	69	00	6C	00	65	00	73	00	OF	00	79	00	00	Af.i.l.e.sy0.
.im	1		1	0x10	0040	00/0	x40	0000	00						<u></u>	0
	610: 00 41 FF 43 5C 41 5C 41 31 46 5C 41 6F 46 5C 41 .img	610: ~ 00 00 41 63 FF FF 43 4F 5C 3D 41 64 FF FF 44 49 5C 3D 41 66 31 00 46 41 5C 3D 41 46 6F 00 46 41 5C 3D 41 66 .img	610: ~ 00 00 00 41 63 00 FF FF FF 43 4F 44 5C 3D 5C 41 64 00 FF FF FF 44 49 52 5C 3D 5C 41 66 00 31 00 30 46 41 54 5C 3D 5C 41 46 00 6F 00 2E 46 41 54 5C 3D 5C 41 66 00 .img	610: ~ 00 00 00 00 41 63 00 6F FF FF FF FF FF 43 4F 44 45 5C 3D 5C 3D 41 64 00 69 FF FF FF FF FF 44 49 52 53 5C 3D 5C 3D 41 66 00 61 31 00 30 00 46 41 54 47 5C 3D 5C 3D 41 46 00 41 6F 00 2E 00 46 41 54 49 5C 3D 5C 3D 41 66 00 69 .img	610: ~ 00 00 00 00 00 00 41 63 00 6F 00 FF FF FF FF FF FF 43 4F 44 45 20 5C 3D 5C 3D 00 41 64 00 69 00 FF FF FF FF FF FF 44 49 52 53 20 5C 3D 5C 3D 00 41 66 00 61 00 31 00 30 00 33 46 41 54 47 45 5C 3D 5C 3D 00 41 46 00 41 00 6F 00 2E 00 74 46 41 54 49 4E 5C 3D 5C 3D 00 41 66 00 69 00 . img0x10	610: ~ 00 00 00 00 00 00 00 41 63 00 6F 00 64 FF FF FF FF FF FF FF 43 4F 44 45 20 20 5C 3D 5C 3D 00 00 41 64 00 69 00 72 FF FF FF FF FF FF FF 44 49 52 53 20 20 5C 3D 5C 3D 00 00 41 66 00 61 00 74 31 00 30 00 33 00 46 41 54 47 45 4E 5C 3D 5C 3D 00 00 41 46 00 41 00 54 6F 00 2E 00 74 00 46 41 54 49 4E 46 5C 3D 5C 3D 00 00 41 66 00 69 00 6C . img0x1000	610: ~ 00 00 00 00 00 00 00 00 41 63 00 6F 00 64 00 FF FF FF FF FF FF FF FF 43 4F 44 45 20 20 20 5C 3D 5C 3D 00 00 B2 41 64 00 69 00 72 00 FF FF FF FF FF FF FF FF 44 49 52 53 20 20 20 5C 3D 5C 3D 00 00 B2 41 66 00 61 00 74 00 31 00 30 00 33 00 2E 46 41 54 47 45 4E 7E 5C 3D 5C 3D 00 00 B2 41 46 00 41 00 54 00 6F 00 2E 00 74 00 78 46 41 54 49 4E 46 4F 5C 3D 5C 3D 00 00 B2 41 66 00 69 00 6C 00	610: ~ 00 00 00 00 00 00 00 00 00 41 63 00 6F 00 64 00 65 FF FF FF FF FF FF FF FF FF 43 4F 44 45 20 20 20 20 5C 3D 5C 3D 00 00 B2 6C 41 64 00 69 00 72 00 73 FF FF FF FF FF FF FF FF FF 44 49 52 53 20 20 20 20 5C 3D 5C 3D 00 00 B2 6C 41 66 00 61 00 74 00 67 31 00 30 00 33 00 2E 00 46 41 54 47 45 4E 7E 31 5C 3D 5C 3D 00 00 B2 6C 41 46 00 41 00 54 00 69 6F 00 2E 00 74 00 78 00 46 41 54 49 4E 46 4F 20 5C 3D 5C 3D 00 00 B2 6C 41 66 00 69 00 6C 00 65 .img0x100400/0x40	610: ~ 00 00 00 00 00 00 00 00 00 00 00 41 63 00 6F 00 64 00 65 00 FF FF FF FF FF FF FF FF FF 43 4F 44 45 20 20 20 20 20 5C 3D 5C 3D 00 00 B2 6C 5C 41 64 00 69 00 72 00 73 00 FF FF FF FF FF FF FF FF FF FF 44 49 52 53 20 20 20 20 20 5C 3D 5C 3D 00 00 B2 6C 5C 41 66 00 61 00 74 00 67 00 31 00 30 00 33 00 2E 00 70 46 41 54 47 45 4E 7E 31 50 5C 3D 5C 3D 00 00 B2 6C 5C 41 46 00 41 00 54 00 69 00 6F 00 2E 00 74 00 78 00 74 46 41 54 49 4E 46 4F 20 54 5C 3D 5C 3D 00 00 B2 6C 5C 41 66 00 69 00 6C 00 65 00 .img0x100400/0x40000	610: ~ 00 00 00 00 00 00 00 00 00 00 00 00 41 63 00 6F 00 64 00 65 00 00 FF FF FF FF FF FF FF FF FF FF 43 4F 44 45 20 20 20 20 20 20 20 5C 3D 5C 3D 00 00 B2 6C 5C 3D 41 64 00 69 00 72 00 73 00 00 FF FF FF FF FF FF FF FF FF FF FF 44 49 52 53 20 20 20 20 20 20 5C 3D 5C 3D 00 00 B2 6C 5C 3D 41 66 00 61 00 74 00 67 00 65 31 00 30 00 33 00 2E 00 70 00 46 41 54 47 45 4E 7E 31 50 44 5C 3D 5C 3D 00 00 B2 6C 5C 3D 41 46 00 41 00 54 00 69 00 6E 6F 00 2E 00 74 00 78 00 74 00 46 41 54 49 4E 46 4F 20 54 58 5C 3D 5C 3D 00 00 B2 6C 5C 3D 41 66 00 69 00 6C 00 65 00 73 . img0x100400/0x400000	610: ~ 00 00 00 00 00 00 00 00 00 00 00 00 00 41 63 00 6F 00 64 00 65 00 00 00 FF FF 00 43 4F 44 45 20 20 20 20 20 20 20 20 5C 3D 5C 3D 00 00 B2 6C 5C 3D 03 41 64 00 69 00 72 00 73 00 00 00 FF FF 00 44 49 52 53 20 20 20 20 20 20 20 20 5C 3D 5C 3D 00 00 B2 6C 5C 3D 07 41 66 00 61 00 74 00 67 00 65 00 31 00 30 00 33 00 2E 00 70 00 00 46 41 54 47 45 4E 7E 31 50 44 46 5C 3D 5C 3D 00 00 B2 6C 5C 3D 11 41 46 00 41 00 54 00 69 00 6E 00 6F 00 2E 00 74 00 78 00 74 00 00 46 41 54 49 4E 46 4F 20 54 58 54 5C 3D 5C 3D 00 00 B2 6C 5C 3D D6 41 66 00 69 00 6C 00 65 00 73 00 .img0x100400/0x400000	610: ~ 00 00 00 00 00 00 00 00 00 00 00 00 00	610: ~ 00 00 00 00 00 00 00 00 00 00 00 00 00	610: ~ 00 00 00 00 6F 00 64 00 65 00 00 00 0F 00 FE FF FF 00 00	610: ~ 00 00 00 00 00 00 00 00 00 00 00 00 00	610: ~ 00 00 00 00 00 00 00 00 00 00 00 00 00

- Does the root directory span more than one cluster? Look up the *next cluster number* in the FAT.
 - Find ThisFATSecNum and ThisFATEntOffset for the current cluster number
 - Go to ThisFATSecNum and read the 32-bit unsigned value starting at offset ThisFATEntOffset
 - FAT will either give you the next cluster number in the directory or the End of Cluster Chain value

Next cluster number of root directory in FAT

00004000	F8	FF	FF	OF	FF	FF	FF	OF	E8	FF	FF	OF	FF	FF	FF	OF	*
00004010	FF	FF	FF	OF													
00004020	FF	FF	FF	OF													
00004030	FF	FF	FF	OF													
00004040	FF	FF	FF	OF	12	00	00	00	13	00	00	00	14	00	00	00	
00004050	15	00	00	00	16	00	00	00	17	00	00	00	18	00	00	00	
00004060	19	00	00	00	1A	00	00	00	1B	00	00	00	1C	00	00	00	
00004070	1D	00	00	00	1E	00	00	00	1F	00	00	00	20	00	00	00	
00004080	21	00	00	00	22	00	00	00	23	00	00	00	24	00	00	00	! "# \$
00004090	25	00	00	00	26	00	00	00	27	00	00	00	28	00	00	00	\$&'(
000040A0	29	00	00	00	2A	00	00	00	2B	00	00	00	2C	00	00	00)*+,
000040B0	2D	00	00	00	2E	00	00	00	2F	00	00	00	30	00	00	00	/0
000040C0	31	00	00	00	32	00	00	00	33	00	00	00	34	00	00	00	1234
000040D0	35	00	00	00	36	00	00	00	37	00	00	00	38	00	00	00	5678
000040E0	39	00	00	00	3A	00	00	00	3B	00	00	00	3C	00	00	00	9:;<
000040F0	ЗD	00	00	00	3E	00	00	00	3F	00	00	00	40	00	00	00	=>?@
00004100	41	00	00	00	42	00	00	00	43	00	00	00	44	00	00	00	ABCD
00004110	45	00	00	00	46	00	00	00	47	00	00	00	48	00	00	00	EFGHO.

Next cluster number of root directory in FAT EoC=0x0FFFFF8 – directory does not go on

FF OF FF FF FF FF OF FF FF FF FF OF FF FF FF FF OF 12 00 00 00 00 00 00 otherv	OF FF FF OF F FF OF FF 00 00 wise this	FF (FF (00 (00 (00 (00 (0F FF 0F FF 0F FF 00 14 00 18 00 10	FF FF 00 00 00	FF 0F FF 0F FF 0F 00 00 00 00 00 00	
FF 0F FF FF FF FF 0F FF FF FF FF 0F 12 00 00 00 00 15 00 00 00 00 00 other 00 00 other	OF F FF OF FF 00 00 wise this	FF (FF (00 (00 (00 (00 (0F FF 0F FF 00 14 00 18 00 1C	FF FF 00 00	FF 0F FF 0F 00 00 00 00 00 00	
FF OF FF FF FF FF 0F 12 00 00 00 00 16 00 00 00 00 00 16 00 00 00 00 00 16 00 00 00 00 00 other	or FF oo oo wise this	FF (00 (00 (00 (00 (0F FF 00 14 00 18 00 1C	FF 00 00 00	FF 0F 00 00 00 00 00 00	
FF 0F 12 00 00 00 00 00 00 00 00 00 00 00 00	wise this	00 (00 14 00 18 00 1C	00 00	00 00 00 00 00 00	
	wise this	00 (00 (00 (00 18 00 1C	00	00 00	
00 00otherv	wise this	00 0	00 1C	00	00 00	
00 00 would	ho tho	00 0	00 20			
		10.00	00 20	00	00 00	
	bethe	00 (00 24	00	00 00) !"
oo oo next c	cluster	00 (00 28	00	00 00	\$&'(
00 00	hor	00 (00 2C	00	00 00))*+,
00 00	Jei	00 (00 30	00	00 00	/0
00 00 32 00 00	00 33 00	00 (00 34	00	00 00	1234
00 00 36 00 00	00 37 00	00 0	00 38	00	00 00	5678
00 00 3A 00 00	00 3B 00	00 0	00 3C	00	00 00	9:,
00 00 3E 00 00	00 3F 00	00 0	00 40	00	00 00	=>@
00 00 42 00 00	00 43 00	00 0	00 44	00	00 00	ABCD
00 00 46 00 00	00 47 00	00 (00 48	00	00 00	EFGHO.
	D0 00 D0 D0 <thd0< th=""> D0 D0 <thd< td=""><td>D0 00 D0 00 D0 <thd0< th=""> D0 D0 D0<!--</td--><td>Description Description <thdescription< th=""> <thdescription< th=""></thdescription<></thdescription<></td><td>00 <td< td=""><td>D0 00 00 2C 00 00<!--</td--><td>D0 00 <th< td=""></th<></td></td></td<></td></thd0<></td></thd<></thd0<>	D0 00 D0 00 D0 D0 <thd0< th=""> D0 D0 D0<!--</td--><td>Description Description <thdescription< th=""> <thdescription< th=""></thdescription<></thdescription<></td><td>00 <td< td=""><td>D0 00 00 2C 00 00<!--</td--><td>D0 00 <th< td=""></th<></td></td></td<></td></thd0<>	Description Description <thdescription< th=""> <thdescription< th=""></thdescription<></thdescription<>	00 00 <td< td=""><td>D0 00 00 2C 00 00<!--</td--><td>D0 00 <th< td=""></th<></td></td></td<>	D0 00 00 2C 00 00 </td <td>D0 00 <th< td=""></th<></td>	D0 00 <th< td=""></th<>

Directory Structure

 Each directory is made up of one or more directory entries that contain

- File name (or sub-directory name)
- Attributes
- First cluster number
 - Cluster number where file or directory in question starts
- More...

Check FAT Directory Structure (page 22) in MS FAT32 File System Spec for details

Finding Files and Directories

- Files and sub-directory entries can be found by going to their *first cluster number*
 - The directory entry for a file or sub-directory contains its *first cluster number, remember?*

Suppose we have read in the root directry and want to find the file 'fatgen103.pdf'

ه∂ user@cop4	610:	~		-					-				-		-		
001003F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00100400	41	63	00	6F	00	64	00	65	00	00	00	OF	00	FE	FF	FF	Ac.o.d.e
00100410	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	00	00	FF	FF	FF	FF	
00100420	43	4F	44	45	20	20	20	20	20	20	20	10	00	64	B2	6C	CODEd.1
00100430	5C	ЗD	5C	ЗD	00	00	B2	6C	5C	ЗD	03	00	00	00	00	00	\=\=1\=
00100440	41	64	00	69	00	72	00	73	00	00	00	OF	00	5D	FF	FF	Ad.i.r.s]
00100450	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	00	00	FF	FF	FF	FF	
00100460	44	49	52	53	20	20	20	20	20	20	20	10	00	64	B2	6C	DIRSd.1
00100470	5C	ЗD	5C	ЗD	00	00	B2	6C	5C	3D	07	00	00	00	00	00	\=\=1\=
00100480	41	66	00	61	00	74	00	67	00	65	00	OF	00	16	6E	00	Af.a.t.g.en.
00100490	31	00	30	00	33	00	2E	00	70	00	00	00	64	00	66	00	1.0.3pd.f.
001004A0	46	41	54	47	45	4E	7E	31	50	44	46	20	00	64	B2	6C	FATGEN~1PDF .d.1
001004B0	5C	ЗD	5C	ЗD	00	00	B2	6C	5C	ЗD	11	00	FD	89	07	00	\=\=1\=
001004C0	41	46	00	41	00	54	00	69	00	6E	00	OF	00	D2	66	00	AF.A.T.i.nf.
001004D0	6F	00	2E	00	74	00	78	00	74	00	00	00	00	00	FF	FF	ot.x.t
001004E0	46	41	54	49	4E	46	4F	20	54	58	54	20	00	64	B2	6C	FATINFO TXT .d.1
001004F0	5C	ЗD	5C	ЗD	00	00	B2	6C	5C	ЗD	D6	03	35	01	00	00	\=\=1\=5
00100500	41	66	00	69	00	6C	00	65	00	73	00	OF	00	79	00	00	Af.i.l.e.sy0.
fat32	.im	1		100	0x10	0040	00/0	0x40	0000	00							0

Suppose we have read in the root directry and want to find the file 'fatgen103.pdf'

0100250	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
irectory e	ontr	v fa	h		00	64	00	65	00	00	00	OF	00	FE	FF	FF	Ac.o.d.e
		y It	51		FF	FF	FF	FF	FF	FF	00	00	FF	FF	FF	FF	
fatgen10)3.p	odf			20	20	20	20	20	20	20	10	00	64	B2	6C	CODEd.1
	~				00	00	B2	6C	5C	ЗD	03	00	00	00	00	00	\=\=1\=
0100440	41	0		69	00	72	00	73	00	00	00	OF	00	5D	FF	FF	Ad.i.r.s]
0100450	FF	FF	E L		FF	FF	FF	FF	FF	FF	00	00	FF	FF	FF	FF	
0100460	44	49	52	53	0	20	20	20	20	20	20	10	00	64	B2	6C	DIRSd.1
0100470	5C	ЗD	5C	ЗD	00	20	B2	6C	5C	ЗD	07	00	00	00	00	00	\=\=1\=
0100480	41	66	00	61	00	74	00	67	00	65	00	OF	00	16	6E	00	Af.a.t.g.en.
0100490	31	00	30	00	33	00	2E	00	70	00	00	00	64	00	66	00	1.0.3pd.f.
01004A0	46	41	54	47	45	4E	7E	31	50	44	46	20	00	64	B2	6C	FATGEN~1PDF .d.1
01004B0	5C	ЗD	5C	ЗD	00	00	B2	6C	5C	ЗD	11	00	FD	89	07	00	\=\=1\=
01004C0	41	46	00	41	00	54	00	69	00	6E	00	OF	00	D2	66	00	AF.A.T.i.nf.
01004D0	6F	00	2E	00	74	00	78	00	74	00	00	00	00	00	FF	FF	ot.x.t
01004E0	46	41	54	49	4E	46	4F	20	54	58	54	20	00	64	B2	6C	FATINFO TXT .d.1
01004F0	5C	3D	5C	3D	00	00	B2	6C	5C	ЗD	D6	03	35	01	00	00	\=\=1\=5
0100500	41	66	00	60	00	60	00	65	00	73	00	OF	00	70	00	00	Afiles v O

Entry's first cluster number 0x000011 = 17

001003F0 00 00 00 00 00 00 00 00 00 00 00 00 00	x			-						-		-			-		~	610:	ser@cop4	🛃 u:
00100400 41 63 00 6F 00 64 00 65 00 00 00 0F 00 FE FF FF Ac.o.d.e 00100410 FF			00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	003F0	0010
00100410 FF		Ac.o.d.e	FF	FF	FE	00	OF	00	00	00	65	00	64	00	6F	00	63	41	00400	0010
00100420 43 4F 44 45 20 20 20 20 20 20 20 20 10 00 64 B2 6C CODE d.1 00100430 5C 3D 5C 3D 00 00 B2 6C 5C 3D 00 00 00 00 00 \=\=1\= 00100440 6C 3D 5C 3D 00 00 B2 6C 5C 3D 00 00 00 5D FF FF Ad.i.r.s]. 00100450 6C 3D 5C 3D 00 00 6C 5C 3D 00 00 00 00 00 00 \=\=1\= 00100460 6C 3D 5C 3D 00 00 00 B2 6C 5C 3D 00 00 00 00 \=\=1\= 00100470 5C 3D 5C 3D 00 00 00 B2 6C 5C 3D 00 00 00 00 \=\=1\= 00100480 41 66 00 61 0 74 00 67 00 65 0F 00 16 6E 00 Af.a.t.g.en. 00100490 31 00 30 00 8 00 2E 00 70 00 01 00 64 00 66 00 1.0.3pd.f. 001004A0 46 41 54 47 43 4E 7E 31 50 44 46 20 00 64 B2 6C FATGEN~1PDF.d.1 001004B0 5C 3D 5C 3D 00 00 B2 6C 5C 3D 11 00 FD 89 07 00 \=\=1\= 001004B0 5C 3D 5C 3D 00 054 00 69 00 6E 00 0F 00 D2 66 00 AF.A.T.i.nf. 001004C0 41 46 00 41 00 54 00 69 00 6E 00 0F 00 D2 66			FF	FF	FF	FF	00	00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	00410	0010
00100430 5C 3D 5C 3D 00 00 B2 6C 5C 3D 03 00 00 00 00 00 \\=\=1\= 00100440 High word F FF Low word 00 5D FF FF Ad.i.r.s]. 00100450 High word F FF Low word 00 64 B2 6C DIRS d.1 00100470 5C 3D 5C 31 0 00 B2 6C 5C 3D 00 00 00 00 \=\=1\= d.1 00100480 41 66 00 61 0 74 00 67 00 65 0F 00 16 6E 00 Af.a.t.g.en. 00100490 31 00 30 00 3 00 2E 00 70 00 01 00 64 00 66 00 1.0.3pd.f. 00100480 46 41 54 47 43 4E 7E 31 50 44 46 20 00 64 B2 6C FATGEN~1PDF .d.1 00100480 5C 3D 5C 3D 00 00 00 00 B2 6C 5C 3D 11 00 FD 89 07 00 \=\=1\= 00100480 5C 3D 5C 3D 00 07 8 00 74 00 69 00 6E 00 0F 00 D2 66 00 AF.A.T.i.nf.		CODEd.1	6C	B2	64	00	10	20	20	20	20	20	20	20	45	44	4F	43	00420	0010
00100440 High word F FF Low word 00 5D FF FF Ad.i.r.s] 00100450 F FF Low word FF FF FF FF Md.i.r.s] 00100460 5C 3D 5C 31 0 00 B2 6C 5C 3D 00 00 00 00 00 \=\=1\= 00100470 5C 3D 5C 31 0 00 B2 6C 5C 3D 00 00 00 00 00 \=\=1\= 00100480 41 66 00 61 0 74 00 67 00 65 0F 00 16 6E 00 Af.a.t.g.en. 00100490 31 00 30 00 3 00 2E 00 70 00 00 00 64 B2 6C FATGEN~1PDF .d.1 001004A0 46 41 54 47 43 4E 7E 31 50 44 46 20 00 64 B2 6C FATGEN~1PDF .d.1 001004B0 5C 3D 5C 3D 00 00 B2 6C 5C 3D 11 00 FD 89 07 00 \=\=1\= 001004B0 5C 3D 5C 3D 00 69 00 6E 00 0F 00 D2 66 00 AF.A.T.i.nf. 001004C0 41 46 00 41 00 54 00 69 00 6E 00 0F 00 D2 66 00 AF.A.T.i.nf.		\=\=1\=	00	00	00	00	00	03	ЗD	5C	6C	B2	00	00	ЗD	5C	ЗD	5C	00430	0010
00100450 High word F FE Low word FF FF FF FF FF FF FF FF Image: Second		Ad.i.r.s]	FF	FF	5D	00						00	2						00440	0010
00100460 00 64 B2 6C DIRS d.1 00100470 5C 3D 5C 31 0 00 B2 6C 5C 3D 00			FF	FF	FF	FF		ord	WC	$\cap W$		FI	F	rd	WO	ah	Hi		00450	0010
00100470 5C 3D 5C 31 0 00 B2 6C 5C 3D 00 00 00 00 00 00 00 00 00 00 $(-+)=1=$ 00100480 41 66 00 61 0 74 00 67 00 65 0F 00 16 6E 00 Af.a.t.g.en. 00100490 31 00 30 00 8 00 2E 00 70 00 00 00 64 00 66 00 1.0.3pd.f. 001004A0 46 41 54 47 43 4E 7E 31 50 44 46 20 00 64 B2 6C FATGEN~1PDF.d.1 001004B0 5C 3D 5C 3D 00 00 B2 6C 5C 3D 11 00 FD 89 07 00 $(=)=1= 001004C0 41 46 00 41 00 54 00 69 00 6E 00 0F 00 D2 66 00 AF.A.T.i.nf. 001004D0 6E 00 2F 00 74 00 78 00 74 00 00 00 00 00 00 FF FF (=, +, +, +, +, +, +, +, +, +, +, +, +, +,$		DIRSd.1	6C	B2	64	00						20	0			911			00460	0010
00100480 41 66 00 61 0 74 00 67 00 65 0F 00 16 6E 00 Af.a.t.g.en. 00100490 31 00 30 00 8 00 2E 00 70 00 00 00 64 00 66 00 1.0.3pd.f. 001004A0 46 41 54 47 45 4E 7E 31 50 44 46 20 00 64 B2 6C FATGEN~1PDF.d.l 001004B0 5C 3D 5C 3D 00 00 B2 6C 5C 3D 11 00 FD 89 07 00 $= =1 =$ 001004C0 41 46 00 41 00 54 00 69 00 6E 00 0F 00 D2 66 00 AF.A.T.i.nf. 001004D0 6E 00 2F 00 74 00 78 00 74 00 00 00 00 FF FF 0 $= x + x + x + x + x + x + x + x + x + x $		\=\=1\=	00	00	00	00	00		ЗD	5C	6C	B2	00	0	31	5C	ЗD	5C	00470	0010
00100490 31 00 30 00 3 00 2E 00 70 00 00 00 64 00 66 00 1.0.3pd.f. 001004A0 46 41 54 47 43 4E 7E 31 50 44 46 20 00 64 B2 6C FATGEN~1PDF.d.l 001004B0 5C 3D 5C 3D 00 00 B2 6C 5C 3D 11 00 FD 89 07 00 \=\=1\= 001004C0 41 46 00 41 00 54 00 69 00 6E 00 0F 00 D2 66 00 AF.A.T.i.nf. 001004D0 6E 00 2F 00 74 00 78 00 74 00 00 00 00 FE FE		Af.a.t.g.en.	00	6E	16	00	OF		65	00	67	00	74	þ	61	00	66	41	00480	0010
001004A0 46 41 54 47 43 4E 7E 31 50 44 46 20 00 64 B2 6C FATGEN~1PDF .d.1 001004B0 5C 3D 5C 3D 00 00 B2 6C 5C 3D 11 00 FD 89 07 00 \=\=1\= 001004C0 41 46 00 54 00 69 00 6E 00 00 00 D2 66 00 AF.A.T.i.nf. 001004D0 6E 00 2E 00 74 00 74 00 00 00 00 00 FE FE 0 + * +		1.0.3pd.f.	00	66	00	64	00	00	00	70	00	2E	00	3	00	30	00	31	00490	0010
001004B0 5C 3D 5C 3D 00 00 B2 6C 5C 3D 11 00 FD 89 07 00 $= \cdot \cdot \cdot \cdot \cdot \cdot$ 001004C0 41 46 00 41 00 54 00 69 00 6E 00 0F 00 D2 66 00 AF.A.T.i.nf. 001004D0 6E 00 2E 00 74 00 78 00 74 00 00 00 00 FE FE $= \cdot \cdot \cdot \cdot \cdot \cdot$		FATGEN~1PDF .d.1	6C	B2	64	00	20	46	44	50	31	7E	4E	45	47	54	41	46	004A0	0010
001004C0 41 46 00 41 00 54 00 69 00 6E 00 0F 00 D2 66 00 AF.A.T.i.nf.		\=\=1\=	00	07	89	FD	00	11	ЗD	5C	6C	B2	00	00	ЗD	5C	3D	5C	004B0	0010
001004D0 6F 00 2F 00 74 00 78 00 74 00 00 00 00 00 FF FF 0 + + +	I	AF.A.T.i.nf.	00	66	D2	00	OF	00	6E	00	69	00	54	00	41	00	46	41	004C0	0010
		ot.x.t	FF	FF	00	00	00	00	00	74	00	78	00	74	00	2E	00	6F	004D0	0010
001004E0 46 41 54 49 4E 46 4F 20 54 58 54 20 00 64 B2 6C FATINFO TXT .d.]		FATINFO TXT .d.1	6C	B2	64	00	20	54	58	54	20	4F	46	4E	49	54	41	46	004E0	0010
001004F0 5C 3D 5C 3D 00 00 B2 6C 5C 3D D6 03 35 01 00 00 \=\=1\=5		\=\=1\=5	00	00	01	35	03	D6	ЗD	5C	6C	B2	00	00	ЗD	5C	3D	5C	004F0	0010
00100500 41 66 00 69 00 6C 00 65 00 73 00 0F 00 79 00 00 Af.i.l.e.sy).	Af.i.l.e.sy0	00	00	79	00	OF	00	73	00	65	00	6C	00	69	00	66	41	00500	0010
fat32.img0x100400/0x4000000)	0							00	00000	x40	00/0	0040	-0x10			1	.im	fat32	

Plug N=17 into FirstSectorofCluster equation, go to that sector...

001021F0	00	00	0.0	00	00	00	0.0	00	0.0	00	00	00	00	00	00	00		
00102200	25	50	44	46	2D	31	2E	34	OA	25	C3	A4	C3	BC	C3	B6	%PDF-1.4.%	
0102210	C3	9F	OA	32	20	30	20	6F	62	6A	OA	3C	3C	2F	4C	65	2 0 obj.< <td></td>	
0102220	6E	67	74	68	20	33	20	30	20	52	2F	46	69	6C	74	65	ngth 3 0 R/Filte	
0102230	72	2F	46	6C	61	74	65	44	65	63	6F	64	65	3E	3E	AO	r/FlateDecode>>.	
0102240	73	74	72	65	61	6D	OA	78	9C	ED	ЗD	C9	8E	1C	В9	72	stream.x=r	
0102250	F7	FA	8A	3A	OF	50	6D	EE	99	04	84	06	7A	35	EO	DB	:.Pmz5	
0102260	D8	02	7C	18	BC	93	FD	C6	82	31	32	30	73	79	BF	6F	120sy.o	
0102270	C6	46	46	56	25	99	A5	D6	3C	65	6B	54	10	C4	CE	28	.FFV% <ekt(< td=""><td></td></ekt(<>	
0102280	92	C1	D8	18	11	24	73	31	77	F6	F8	8F	СЗ	EF	47	73	Şs1wGs	
0102290	3C	99	72	99	AC	2B	E5	94	A1	FC	E3	EF	C7	FF	FC	E9	<.r+	
01022A0	F8	7F	87	9F	8F	BF	1F	33	56	96	D2	1F	83	4F	77	F9		
01022B0	68	A7	BB	54	5B	98	23	FC	FB	E3	7F	OE	73	86	DE	66	hT[.#sf	H
01022C0	BE	8B	C7	CF	07	1B	7C	69	43	DO	6F	47	9B	E7	3B	57	iC.oG;W	
01022D0	21	17	DD	DD	54	21	6F	A6	82	98	A1	83	4F	F6	6E	AE	!T!o0.n.	
01022E0	75	C1	A6	BB	DO	AO	C9	94	C1	05	8A	2E	AA	11	08	4A	uJ	
01022F0	D9	96	F2	B7	03	B5	64	88	B1	08	44	23	54	08	47	17	dD#T.G.	

Does the file continue after this cluster?
 Look up current cluster number 17 in FAT...

子 user@cop4	510: -	~									С	Cc lust	ontir ter (nue Dx1	es t 2=	:0 :18!	
00004000	F8	FF	FF	OF	FF	FF	FF	OF	E8	FF	E F		s E	F.F.	F.F.	OF	
00004010	FF	FF	FF	OF	FF	FF	FF	OF	FF	FF		JF	FF	FF	FF	OF	
00004020	FF	FF	FF	OF	FF	FF	FF	OF	FF		FF	OF	FF	FF	FF	OF	
00004030	FF	FF	FF	OF	FF	FF	FF	OF	1	FF	FF	OF	FF	FF	FF	OF	
00004040	FF	FF	FF	OF	12	00	00	00	13	00	00	00	14	00	00	00	
00004050	15	00	00	00	16	00	00	00	17	00	00	00	18	00	00	00	
00004060	19	00	00	00	1A	00	00	00	1B	00	00	00	10	00	00	00	
00004070	1D	00	00	00	1E	00	00	00	1F	00	00	00	20	00	00	00	
00004080	21	00	00	00	22	00	00	00	23	00	00	00	24	00	00	00	! "# \$ –
00004090	25	00	00	00	26	00	00	00	27	00	00	00	28	00	00	00	\$8
000040A0	29	00	00	00	2A	00	00	00	2B	00	00	00	2C	00	00	00) * + ,
000040B0	2D	00	00	00	2E	00	00	00	2F	00	00	00	30	00	00	00	/0
000040C0	31	00	00	00	32	00	00	00	33	00	00	00	34	00	00	00	1234
000040D0	35	00	00	00	36	00	00	00	37	00	00	00	38	00	00	00	5678
000040E0	39	00	00	00	3A	00	00	00	3B	00	00	00	3C	00	00	00	9;<
000040F0	3D	00	00	00	3E	00	00	00	3F	00	00	00	40	00	00	00	=>?@
00004100	41	00	00	00	42	00	00	00	43	00	00	00	44	00	00	00	ABCD
00004110	45	00	00	00	46	00	00	00	47	00	00	00	48	00	00	00	EFGHO.
fat32	.ima	r			-0x40	008	/0x4	1000	000-								0.

Summary of Finding Files/Dirs

- Find *first cluster number* in directory entry of the file or directory at hand
- Figure out the sector to read using cluster number and FirstSectorofCluster equation
- Read that cluster
- Figure out if the file or directory continues past cluster by looking up FAT[current cluster number]
 - If EoC mark stop
 - Else go to 3 with cluster=FAT[current cluster number]

To Do

- Write code to parse the Boot Sector. Get the necessary values, print them and check.
- Access the Boot Directory. Get the list of files and folders. Print them and check.
- Open a particular file and read from it. Use FAT Table info to get all the clusters associated with it.

Next Time

- Discussion of specific file operations (For example: writing to files, creating and deleting files and directories etc.)
- More discussion of directory entries