
Recitation 3

— Welcome, everybody! —

Topics Covered

- FAT File System
- FAT vs Inode
- Hexdump Guide (PennOS preview!)

FAT File System

How can I edit a file on a FAT based File System?

1. Does the file exist? Am I allowed to access it? Where can I find the file?
2. Where is the n^{th} byte of the file? That's where I want to begin editing.
3. Access the file via some block math.

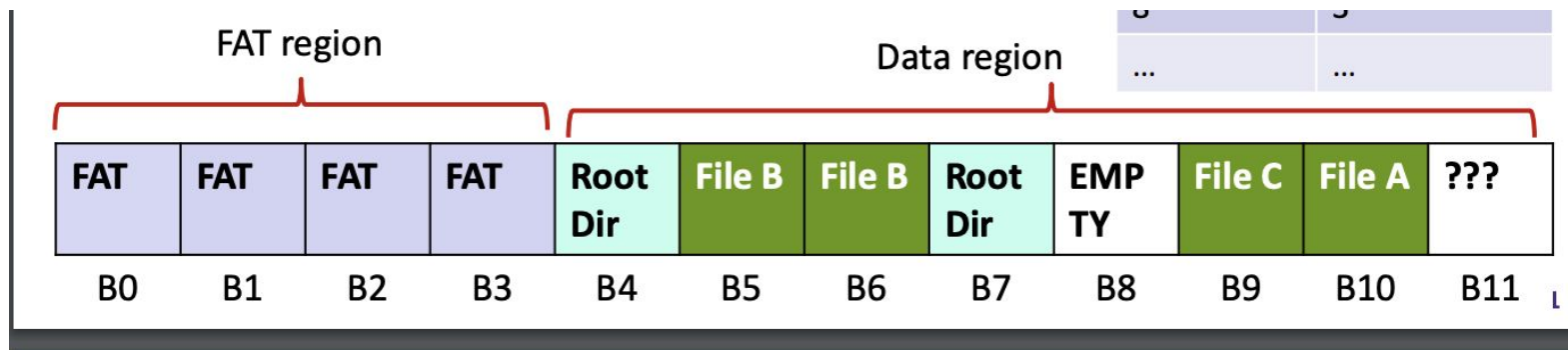


File Allocation Table

FAT system splits to two parts:

FAT region & Data Region

Data Region has *File Contents & Directory Contents*



Does the file exist? Can I access it? Where is the file?

The Root Directory starts at a *known* spot within the Data Region

Root directory stores info about other files.

1. Search by **name**
2. Verify **permissions**
3. Return **firstBlock**

Root Directory Entry (PennFAT)

```
char name[32];  
uint32_t size;  
uint16_t firstBlock;  
uint8_t type;  
uint8_t perm;  
time_t mtime;  
// The remaining 16 bytes are reserved
```

Navigating the FAT

FAT[current] = next

Each entry is 2 bytes (0xABCD)

[0] # of FAT blocks (MSB) & Block Size (LSB)

[1] **ROOT DIRECTORY.** First data block is the root dir.

(FAT[2].....FAT[MSB]) are file (or extended root directory) block numbers.

Index	Link
0	0x2004 ← MSB=0x20 (32 blocks in FAT), LSB=0x04 (4K-byte block size)
1	0xFFFF ← Block 1 is the only block in the root directory file
2	5 ← File A starts with Block 2 followed by Block 5
3	4 ← File B starts with Block 3 followed by Block 4
4	0xFFFF ← last block of File B
5	6 ← File A continues to Block 6
6	0xFFFF ← last block of File A
...	...

Finding the block address

e.g. **Block Size** = 4 KB; **# of Blocks in FAT** = 32

File: "Foo" has firstBlock at 2. I want to edit the 9,021 B

"Foo's Block" = $\text{ceil}(9,021 \text{ B} / 4 \text{ KB}) = 3^{\text{rd}}$

Block Address = (Blocks in Fat) (Block Size) + ("Foo's Block" - 1) (Block Size)

Skip FAT Region 1-indexing

Block Offset = $9,021 \% \text{ Block Size} = 829$

Desired Byte Location = Block Address + Block Offset

= 164,669 "Bytes" from the start of the file system

Block #	Next
0	BITMAP/SPECIAL
1	END
2	6
3	9
4	END
5	EMPTY / UNUSED
6	3
7	END
8	END
9	END
10	8
11	END

INodes

Why use I-Nodes?

Recap: What is an Inode?

- An Inode (Index Node) is a structure that stores metadata about a file (like permissions, size, and owner) and **the locations of the file's data blocks**. Instead of requiring all blocks to be continuous, the inode points to wherever the blocks are scattered on the disk.

Feature	Contiguous Allocation	Inode-Based Allocation
Data Block Layout	All blocks for a file are stored sequentially in one continuous chunk on the disk.	Data blocks for a file can be scattered anywhere on the disk. The inode links them together.
Sequential Read Speed	Very Fast. Minimal disk head movement is required since all data is physically next to each other.	Slower. The disk head may need to jump to different locations to read all the blocks.
File Growth	Hard. If a file grows, a larger contiguous block of free space must be found, and the file must be copied over.	Easy. New blocks can be allocated from anywhere on the disk and simply added to the inode's list of pointers.
Metadata Storage	Typically stored in the directory entry.	Stored in the inode itself, separate from the directory entry.

Key Takeaways

1. **Contiguous Allocation:** Think **performance and simplicity** for files that don't change size. It excels at fast sequential reads but struggles with file growth and disk fragmentation.
2. **Inode-Based Allocation:** Think **flexibility and efficiency** for dynamic files. It handles file size changes and fragmentation gracefully at the cost of some performance overhead for sequential reads.

Comparison Question Pt 1)

Question: Which file allocation scheme is best for small, fixed-size files that are frequently read but sometimes written to: contiguous or Inodes? **Why?**

Answer:

- **Contiguous** because the size of the files stays the same, so we **do not need to reallocate**. In this case, we can specifically use **caching** to take advantage of **spatial locality** since we will never have to move existing files around given that they have constant size.
- Inodes would introduce unnecessary overhead for these specific use cases.

Comparison Question Pt. 2)

Question: If we need to store many very large files that stretch several blocks and are frequently modified including file extending/shortening and rearrangement, which allocation scheme is better: contiguous or Inodes? Why?

Answer:

- **Inode-based allocation** is better here because it is better at dealing with **dynamic file sizes** and **fragmentation**. The file's data blocks can be scattered across the disk, which allows efficient modification, extension, and truncation without needing to move the entire file.
- Contiguous allocation would **not** work here because frequent length modifications would require reallocation to find a large enough block of free space.

Hexdump Example

What is a HexDump?

- A hexdump is a representation of binary data in hexadecimal format.
- The `hd` command shows the raw contents of a file or disk, byte by byte. We will use this a lot in Penn OS for the file system to check our progress and debug
- A hexdump has three parts: the offset, the hex data, and ASCII representation.

What does a Hexdump look like?

F A T D A T A	vagrant@cis548Dev:~\$ hd minfs		
	00000000	00 01 ff ff ff ff 00 00 00 00 00 00 00 00 00 00	
	00000010	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
	*		
	00000100	66 31 00 00 00 00 00 00 00 00 00 00 00 00 00 00	f1.....
	00000110	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
	00000120	00 01 00 00 02 00 01 06 b6 1e 83 60 00 00 00 00	
	00000130	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
	*		
	00000200	68 65 6c 6c 6f 00 56 ed b8 e7 6d 3d c8 a6 cd f5	hello.....
	00000110	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
	*		
	00008000		
Offset		Hex Data	ASCII Rep.

We will dive into the specifics next, but for some context, this is a dump of a file system that has 1 FAT block, a block size of 256B, and 1 file (f1) that just contains the word "hello".

Little Endian Explained

- Little-endian is a byte-ordering scheme where the **least significant byte comes first**. This is a common point of confusion we see when students read hexdumps.
- In the minfs FAT, 00 01 is read as 0x0100. The computer interprets the bytes in reverse order to form the full value. Which in this case is # of blocks in Fat, Block Size
- When we see the first block of f1 is 02 00, this means the first block is #2.

Diving into our Hexdump

- Let's look at the *minfs* hexdump which we use in PennOS.
- When we run `mkfs minfs 1 0`, it creates an empty file system that has **1 block in FAT** and a **block size of 256B**.
- Our first block is the FAT and our other 127 ($256/2 - 1$) blocks are reserved for Data
- The first two bytes of the FAT, `FAT[0]`, are **important** because they give us the block size and # of blocks, which is essential for navigating our file system.

Minfs Hexdump

```
vagrant@cis548Dev:~$ pennfat
pennfat# mkfs minfs 1 0
vagrant@cis548Dev:~$ ls -l minfs maxfs
-rw-r--r-- 1 vagrant vagrant 268558336 Apr 23 19:21 maxfs
vagrant@cis548Dev:~$ hd minfs
00000000 00 01 ff ff 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00008000
vagrant@cis548Dev:~$ cp minfs minfs.1
```

- `FAT[0] = 00 01`, which represents `0x0100` (**due to little-endian byte order**). This signifies a 1-block FAT and a block size of 256 bytes.
- `FAT[1] = ff ff`, which represents the File System ending at block 1.
 - Note: The File System always starts at block 1

Hexdump with a File Added

- Let's follow the file f1 through our hexdump. When we add f1, the file system updates the **FAT** and the **Directory Entry**.
- The **FAT** is updated to link the file's data blocks. Since f1's first block is #2, we see that the FAT entry for block #2 is ff ff, which means end of file.
- The directory entry is located at offset 00000100, which is the start of the root directory.
 - **66 31 00...**: This is the filename, "f1", padded with null bytes.
 - **00 01**: These bytes represent the size of the file, 0x0100, which is 1 block.
 - **02 00**: These bytes indicate the first block address of the file, 0x0002 which is block #2
 - **00 01 06 b6 1e 83 60 00 00 00 00 00 00 00 00**: This is the remaining metadata, including a timestamp and permissions.
- Finally, at offset 00000200 (block #1), we find the actual file data.
 - **68 65 6c 6c 6f 00**: These are the ASCII codes for "hello" followed by a null terminator. The ASCII representation on the far right confirms this.

File System with 1 File Added

```
vagrant@cis548Dev:~$ pennfat
pennfat# mount minfs
pennfat# touch f1
pennfat# cat -a f1
hello
pennfat# ls
  2 -rw-   6 Apr 23 19:23:34 2021 f1
pennfat# cp f1 -h demo2copy
pennfat#
vagrant@cis548Dev:~$ cmp demo2 demo2copy
vagrant@cis548Dev:~$ hd minfs
```

F A T	00000000	00 01 ff ff	ff ff	00 00 00 00 00 00 00 00 00 00 00 00 00 00	
	00000010	00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00		
	*				
R O O T	00000100	66 31 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00	f1.....	
	00000110	00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00		
	00000120	00 01 00 00	02 00 01 06 b6 1e 83 60 00 00 00 00 00 00		
	00000130	00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00		
	*				
F 1	00000200	68 65 6c 6c 6f 00 56 ed b8 e7 6d 3d c8 a6 cd f5	hello.....		
	00000110	00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00		
	*				

```
00008000
```

```
vagrant@cis548Dev:~$ cp minfs minfs.2
```

Hexdump Navigation

1. What is the index of f1's second block?

5

File: recitation_3_hd	
1	00000000 00 01 ff ff 00 04 00 ff ff ff 00 00 00 00
2	00000010 00 00 00 00 00 00 00 00 00 00 00 00 00
3	*
4	00000100 66 31 00 00 00 00 00 00 00 00 00 00 00 f1.....
5	00000110 00 00 00 00 00 00 00 00 00 00 00 00 00
6	00000120 00 02 00 00 02 00 00 00 b5 1e c7 68 00 00 00 h..
7	00000130 00 00 00 00 00 00 00 00 00 00 00 00 00
8	00000140 66 32 00 00 00 00 00 00 00 00 00 00 00 f2.....
9	00000150 00 00 00 00 00 00 00 00 00 00 00 00 00
10	00000160 00 02 00 00 03 00 00 06 b5 1e c7 68 00 00 00 h..
11	00000170 00 00 00 00 00 00 00 00 00 00 00 00 00
12	*
13	00000200 9b 3b 26 43 82 a8 68 80 be 63 bd 2b bb b8 d9 ad .8AC..h..c.+...
14	00000210 c8 87 1b 9a a9 3d 67 e9 75 cf c3 76 d4 1e 84 dc ...g.u.v..
15	00000220 f2 e2 dc 3e cc a6 51 e3 ad ef 1b 7e 87 7c 56 46 >...Q...-[VF
16	00000230 a0 8d c8 72 d1 86 c6 1a f7 f5 d6 69 55 b1 9c 40 r...IU..0
17	00000240 65 64 5a 73 30 ef dc c5 76 d5 aa d8 93 db 66 b6 e.Z88..v...f.
18	00000250 c6 b6 ab 24 1d 2f f2 22 1c 4d 59 d9 af 4e f1 13 ...\$/..*MY..N..
19	00000260 20 38 83 fc c7 a5 ab 00 5e 7d 95 13 fa 1c db b8 0.....*}
20	00000270 c4 a8 c3 74 a3 7a 88 00 8a 9f 4f e7 1e 40 32 53 ...t.z...0..02S
21	00000280 55 fd 88 06 89 c6 ae a0 81 ca fc 08 4a 39 92 9a U.....39..
22	00000290 49 41 46 6f 40 00 00 b3 42 80 3c 0c 04 a4 2d ...F08..B..0..
23	000002a0 51 37 81 c6 88 43 f2 d3 d2 61 56 96 33 7e a3 07....C...aV3e..
24	000002b0 44 17 f7 14 8a 6b 9d 03 55 81 3f ef 53 d3 b9 bd D....K.U.7.S..
25	000002c0 fc e5 9b b5 6e a9 d8 32 3e c4 5f 9e 12 2d 85 20 ...n..2>....
26	000002d0 f3 43 14 33 c4 24 40 7c 2f c2 66 b1 06 59 08 74 .CA3.S8[.1.V.y.t
27	000002e0 a9 bd b2 8e 05 f0 3f af 85 45 e9 b4 67 8c b6 H...E.g..
28	000002f0 ba 0e 6a fb cc 8d 8a fd f0 c5 3a 5a 5f 70 c4 92 ...n.....AT...
29	00000300 d2 9e 5f 47 27 20 e5 bd 70 a0 84 5a 4d ce 56 22 ...G'...p.Zm.V^
30	00000310 c9 9c 46 59 51 fe 8d 35 86 f6 2e 78 0f b1 db 07 ...FYQ..5...x...
31	00000320 7b 50 d9 ee 5c d2 f3 ac 96 79 c6 cd 3b 9c 1e e2 P...V...y..j...
32	00000330 cd c2 c2 7e 5f ff 47 f7 b3 fa 87 9a 4e fb fa 01 ...G.....N...
33	00000340 b7 d2 5d 08 f4 d7 b7 5f 64 84 71 9f e6 d0 10 ...1.....m.q..J..
34	00000350 7a dd e3 b7 33 fe 56 c5 31 92 44 44 7e a6 bd ac 2M..3.V.1.DD...e
35	00000360 88 a4 4e a9 6a 88 d8 fb 29 82 35 70 9a ad 43 16 ...N..j...)5p.MC.
36	00000370 86 75 1d 18 15 c0 a5 c2 ac f8 4a ed dd 55 92 ba U.....U...
37	00000380 cd 46 c3 3a c3 7a 43 bc e2 ca 16 08 8f 61 86 f1 .F.:2C...P.a...
38	00000390 6f 40 86 71 40 fa e6 27 5e 7a 3b ef c0 d4 14 49 00008...c..t8...I
39	000003a0 8c 8b 4f c9 f0 a0 54 db a2 9f aa 17 65 f4 39 54 ...O...T.....e9T
40	000003b0 87 9d 15 21 12 5f 21 b7 92 46 81 3a 10 4a b2 98 ...1...F.:J...
41	000003c0 0e 9e aa ad a3 3d 18 bc 8c 18 be 2c a9 79 b4 f6 y..j...
42	000003d0 08 83 fe ed b1 be 4b af 7d 5c 71 53 81 15 5a 89 K.)qS..Z...
43	000003e0 c4 83 7e 8e 9a d9 64 8a 2a 21 29 ab 6c e1 3b 54 ...-...d..1..jT
44	000003f0 b9 38 66 44 27 24 c7 ab 01 11 71 6e 12 f3 4a ce ...0.D'\$...qm..J..
45	00000400 2a a5 0b 5e 72 bd a6 85 14 14 83 4c 11 99 c8 4a *..*r.....L...J
46	00000410 9f 55 07 ca e2 7f a5 e5 23 2a fd 49 6d c5 88 b8 U.....*..Im..
47	00000420 85 33 b8 1b 61 21 29 0e e5 1a 8c d9 66 07 aa {..3...a1).....g.
48	00000430 8d 05 92 17 c7 08 c5 46 69 61 a8 08 76 f9 ee M.....h.F.a..v..
49	00000440 0a 09 d5 e9 25 89 ac 6d 00 4d 23 70 ba 9c da a2 *...000...j
50	00000450 65 c2 a0 6f 9b a0 ae 64 1c 5e 70 bd 3e b7 41 6d e..o...d.'p>..Am
51	00000460 b3 25 d0 e3 38 c1 58 4c 12 44 dd a0 e5 ad 8c b8 ...N..8.XL.D.....
52	00000470 ac d1 a2 a5 94 3d 45 a5 14 8e d9 41 4b 87 8f ae E.....AK..N
53	00000480 55 1b 8b 8f 1a 08 78 86 c3 1a 0e 3a 81 b1 1b 27 2e...x...K..V'
54	00000490 53 72 31 cc af 23 89 a5 b1 29 cb 01 fc 81 66 1b S1L...-...V..
55	000004a0 c8 9a f2 95 3c 3c 82 a8 7a 1c 7f 8f 60 67 f7 82 <<2.....
56	000004b0 66 1c 44 45 35 5c 47 a4 47 03 04 64 01 e8 b5 c4 f.DES(G.G..d....
57	000004c0 d0 40 0f 10 e7 d5 e5 e4 79 a5 24 bb 75 4d 6e 3a {e.....p\$.uMn:
58	000004d0 33 07 83 99 95 2b 75 6a 56 15 aa db be 63 2d 7c 3...MuV.....
59	000004e0 e9 c8 52 93 cc 96 e5 6d 1a ff 24 16 07 0e 87 68 ...R.....
60	000004f0 64 a4 6d 3f 07 a2 6c 72 3d 1d 10 37 9d 55 3c m.mP...lrm-7..uK
61	00000500 c6 34 12 2e 58 25 73 64 1d 96 19 65 19 81 7d 05 ...4..Xsd...e..}
62	00000510 30 b1 83 ff ce 7c 43 6c 4a 1d af 04 76 ac d5 20 0....[C1]...v..
63	00000520 02 e6 62 b5 28 c7 bd b6 23 13 d7 e7 6a 88 00 93 ...b.(...#.j...
64	00000530 73 cf 24 06 d8 ce 6d 64 ba 80 fe ea 2f 08 1c c1 a.S...nd.../...
65	00000540 4f 9a 33 9b 9b 66 d8 a0 ae 8a 99 32 ee 8c 61 c1 0.3..f.....2....
66	00000550 22 08 d0 9d 0f 9a 31 ae ef 51 44 bb 74 72 db 29 *....1..0D.tr.)
67	00000560 d7 61 f4 89 71 f0 b4 9e 82 99 35 8e 0b dc 34 e3 ...a.q.....5..4.
68	00000570 c3 e8 83 f9 c8 5e b2 15 1e af 4a 2c 65 b1 36 8e e.5..
69	00000580 07 87 17 af 02 4a bf 82 b7 3c 62 04 fa ab a2 ...N.....0...e..
70	00000590 6d 2a 0a ac 35 4e f5 fd 24 72 95 cd 23 85 7a 9e m...8a..Sr...e..
71	000005a0 c7 22 65 39 f8 97 7e c8 bc 9a 15 37 95 22 ca 00 ...e9...-7..*..
72	000005b0 ab d0 61 7e 42 4c f1 34 63 1b ce 77 94 16 9c 66 ...a-BL.4c..w...f
73	000005c0 ec 06 34 4f f4 42 b7 8e 51 49 7a 60 42 08 de e8 ...40.B..Q1zK8...
74	000005d0 73 09 c3 50 cb 3b 08 fa ee 4a c3 5c 09 d7 1f).P...j...3...
75	000005e0 59 b6 87 01 83 87 87 43 68 e1 5a b9 44 05 17 e4 Y.....Z.D...
76	000005f0 7e df e8 bd 84 1b 74 03 78 45 c3 68 1f 06 c5 b7 t.xE.h....
77	00000600 00 00 00 00 00 00 00 00 00 00 00 00 00 00
78	*
79	00000000
80	

On Gradescope

Hexdump Navigation

2. What is the block size of the hexdump?

256

File: recitation_3_hd	
1	00000000 00 01 ff ff 05 00 04 00 ff ff ff ff 00 00 00 00
2	00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00
3	*
4	00000100 66 31 00 00 00 00 00 00 00 00 00 00 00 00 f1.....
5	00000110 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6	00000120 00 02 00 00 02 00 00 00 b5 1e c7 68 00 00 00 h..
7	00000130 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8	00000140 66 32 00 00 00 00 00 00 00 00 00 00 00 00 f2.....
9	00000150 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10	00000160 00 02 00 00 03 00 00 00 b5 1e c7 68 00 00 00 00 h..
11	00000170 00 00 00 00 00 00 00 00 00 00 00 00 00 00
12	*
13	00000200 9b 3b 26 43 82 a8 68 00 be 63 bd 2b bb b8 d9 ad .8AC..h..c+.
14	00000210 c8 87 1b 9a a9 3d 67 e9 75 cf c3 76 d4 1e 84 dc ...g.u.v..
15	00000220 f2 e2 dc 3e cc a6 51 e3 ad ef 1b 7e 87 7c 56 46 >...Q...-[VF
16	00000230 a0 8d c8 72 d1 86 c6 1a f7 f5 d6 69 55 b1 9c 40 r... U.. 0
17	00000240 65 64 5a 73 30 ef dc c5 76 d5 aa d8 93 db 66 b6 e.Z88..v...f..
18	00000250 c6 b6 ab 24 1d 2f f2 22 1c 4d 59 d9 af 4e f1 13 ...\$/..*MY..N..
19	00000260 20 38 83 fc c7 a5 ab 00 5e 7d 95 13 fa 1c db b8 0.....*}
20	00000270 c4 a8 c3 74 a3 7a 88 00 8a 9f 4f e7 1e 40 32 53 ...t.z...0..02S
21	00000280 55 fd 88 06 89 c6 ae a0 81 ca fc 08 4a 39 92 9a U.....39..
22	00000290 d4 09 41 46 6f 40 00 0b b3 42 80 3c 4e ca 2d ...eF08..B..-
23	000002a0 51 37 81 c8 eb 88 43 f2 d3 d2 61 56 96 33 7e a3 07....C...aV3e..
24	000002b0 44 17 f7 14 8a 6b 9d 03 55 81 3f ef 53 d3 b9 bd D....K..U.7.S..
25	000002c0 fc e5 98 b5 6e a9 d8 32 3e c4 5f 9e 12 2d 85 20 ...n..2>....
26	000002d0 f3 f3 41 33 c4 24 40 7c 2f c2 66 b1 06 59 08 74 c.A3.S8[.l.V.y.t
27	000002e0 a9 bd b2 0e 05 f0 3f af 85 45 e9 b4 67 8c b6 H...E.g..
28	000002f0 ba 0e 6a fb cc 8d 8a fd f0 c5 3a 5a 5f 70 c4 92 ...n.....AT...
29	00000300 d2 9e 5f 47 27 20 e5 bd 70 a0 84 5a 6d ce 56 22 ...G'...p.Zm.V^
30	00000310 c9 9c 46 59 51 fe 8d 35 86 f6 2e 78 0f b1 db 07 ...FYQ..5...x...
31	00000320 7b 50 d9 0e 5c d2 f3 ac 96 79 c6 cd 3b 9c 1e e2 {P..\\...y..j...
32	00000330 bd c2 7e 5f ff 47 f7 b3 fa 87 90 4e fb fa 01 G.....N..
33	00000340 b7 d2 5d 08 f4 d7 01 5a 64 84 71 9f c6 10 00 e..l...m..q..j..
34	00000350 7a dd e3 b7 33 fe 56 c5 31 92 44 44 7e 6e bd ac 2M..3.V.I.DD-..
35	00000360 88 a4 4e a9 6a 88 d8 fb 29 82 35 70 9a ad 43 16 ..N.j..).5p.MC..
36	00000370 86 75 1d 18 15 c0 a5 c2 ac f8 4a ed dd 55 92 ba U.....U...
37	00000380 cd 46 c3 3a c3 7a 43 bc e2 ca 16 08 8f 61 86 f1 f..ZC...P.a..
38	00000390 6f 40 86 71 40 fa e6 27 5e 74 3b ef c0 d4 14 49 00009..B...t8..
39	000003a0 8c 8b 4f c9 f0 a0 54 db a2 9f aa 17 65 f4 39 54 ...O...T.....e9T
40	000003b0 87 9d 15 21 12 5f 21 b7 92 46 81 3a 10 4a b2 98 ...l...f..:J...
41	000003c0 0e e9 aa ad a3 3d 18 bc 8c 18 be 2c a9 79 b4 f6 y...
42	000003d0 08 83 fe ed b1 be 4b af 7d 5c 71 53 81 15 5a 89 K.)qS..Z..
43	000003e0 c4 83 7e 8e 9a d9 64 8a 2a 21 29 eb 6c e1 3b 54 ...-...d..).l;J
44	000003f0 b9 38 66 44 27 24 c7 ab 01 11 71 6e 12 f3 4a ce ...0.D'\$...qm..j..
45	00000400 2a a5 0b 5e 72 bd a6 85 14 14 83 4c 11 99 c8 4a *..*r.....l...J
46	00000410 9f 55 07 ca e2 7f a5 e5 23 2a fd 49 6d c5 88 b8 U.....#..Im..
47	00000420 85 33 b8 1a 1b 61 21 29 0e e5 1a 8c d9 66 67 aa {3...a1).....g..
48	00000430 8d 05 92 19 c7 08 c5 45 69 61 a8 08 76 f9 ae M.....h.f.a..v..
49	00000440 0a e9 d5 e9 25 89 ac 6d 00 4d 23 70 ba 9c da a2 @00-...
50	00000450 65 c2 a0 6f 9b a0 ae 64 1c 5e 70 bd 3e b7 41 6d e..o...d.'p>..Am
51	00000460 b3 25 d0 e3 38 c1 58 4c 12 44 dd a0 e5 ad 8c b8 ..N..8.XL.D.....
52	00000470 ac d1 a2 a5 94 3d 45 a5 14 8e d9 41 4b 87 8f ae E.....AK..N
53	00000480 55 1b 8b 8f 1a 08 78 86 c3 1a eb 3a 81 b1 1b 27 2e....x...K..'
54	00000490 53 72 31 cc af 23 89 a5 b1 29 cb 01 fc 81 66 1b Srl..f...).V..
55	000004a0 c8 9a f2 95 3c 3c 82 a8 7a 1c 7f 8f 60 67 f7 82 <<2.....
56	000004b0 66 1c 44 45 35 5c 47 a4 47 83 04 64 01 e8 b5 c4 f.DES{G.G..d...
57	000004c0 d0 48 0f 10 e7 d5 e5 e4 79 a5 24 bb 75 ad 6e 3a {0.....p\$.uMn:
58	000004d0 33 b7 83 99 95 2b 75 6a 56 15 aa db be 63 2d 7c 3...NuJV.....
59	000004e0 e9 c8 52 93 cc 96 e5 6d 1a ff 24 16 87 0e 87 68 ...R.....
60	000004f0 6d a4 6d 3f 07 a2 6c 72 3d 2d 1d 37 9f 55 3e m.mP..lrm-7..Uk
61	00000500 c6 34 12 2e 58 25 73 64 1d 96 19 65 19 81 7d 05 ...4..Xmsd...e..}
62	00000510 3b b1 83 ff ce 7c 43 6c 4a 1d af 04 76 ac d5 20 0....[Clj...v..
63	00000520 02 e6 62 b5 28 c7 bd b6 23 13 d7 e7 6a 88 00 93 ...b.(...#.j...
64	00000530 73 cf 24 06 d8 ce 6d 64 ba 80 f6 ea 2f 08 1c c1 a\$....nd.../..
65	00000540 4f 9a 33 9b 9b 66 d8 a0 ae 8a 99 32 ee 8c b1 c0 0.3..f...2.....
66	00000550 22 08 db 9d 0f 9a 31 ae ef 51 44 bb 74 72 db 29 *....1..0D.tr.)
67	00000560 d7 61 f4 89 71 f0 b4 9e 82 99 35 8e 0b dc 34 e3 ...a.q....5...4..
68	00000570 c3 e8 83 f9 c8 5e b2 15 1e af 4a 2c 65 b1 36 8e e.5..
69	00000580 07 87 17 1f 02 4a bf 82 b7 3c 62 04 fa ab a2 ...N.....-G...
70	00000590 6d 2a 0a ac 35 4e f5 fd 24 72 95 cd 23 85 7a 9e m..8h..Sr...
71	000005a0 c7 22 65 39 f8 97 7e c8 bc 9a 15 37 95 22 ca 00 ...e9...-7..*..
72	000005b0 ab d0 61 7e 42 4c f1 34 63 1b ce 77 94 16 9c 66 ...a-BL.4c..w...f
73	000005c0 ec 06 3a 4f f4 42 b7 8e 51 49 7a 60 42 08 de e8 ...40.B..QizX8..
74	000005d0 73 29 0c 5f 50 cb 3b 08 fa ee 4a c8 5c 09 d7 1f e).P..;..j...
75	000005e0 59 b6 87 01 83 87 87 43 68 e1 5a b9 44 05 17 e4 Y.....Z.D...
76	000005f0 7e df e8 bd 84 1b 74 03 78 45 c3 68 1f 06 c5 b7 t.xE.h...
77	00000600 00 00 00 00 00 00 00 00 00 00 00 00 00 00
78	*
79	00000000
80	

3. What is the 404th byte of f2?

Desired Block = $\text{ceil}(404/256) = 2^{\text{nd}} \Rightarrow \text{Block 4}$

$$\begin{aligned} \text{Byte Location} &= (\text{Blocks in Fat}) (\text{Block Size}) + (\text{Desired Block} - 1) (\text{Block Size}) + \text{Block Offset} \\ &= (1)(256) + (4-1)(256) + 148 \end{aligned}$$

0x00000494 ⇒ 23

Hexdump Navigation

3. What is the 404th byte of f2?

0x494

		4x1	4x0	4x3	4x2	4x5	4x4													
45	00000400	2a	a5	0b	5e	72	bd	a6	85	14	14	83	4c	11	99	c8	4a	*..^r.....L...J		
46	00000410	9f	55	07	ca	e2	7f	a5	e5	23	2a	fd	49	6d	c5	88	8b	.U.....*#.Im...		
47	00000420	85	33	b0	8a	1b	61	21	29	0e	e5	1a	8c	d9	e6	67	aa	.3...a!).....g.		
48	00000430	4d	8d	a5	92	19	c7	68	c5	46	89	61	a0	08	76	f9	ee	M.....h.F.a..v..		
49	00000440	0a	e9	d5	e9	25	80	ac	d6	00	40	23	70	ba	9c	da	a2	%....@#p....		
50	00000450	65	c2	a0	6f	9b	a0	ae	64	1c	5e	70	bd	3e	b7	41	6d	e..o...d.^p.>.Am		
51	00000460	b3	25	d0	e3	38	c1	58	4c	12	44	dd	a0	e5	ad	8c	b8	.%..8.XL.D.....		
52	00000470	ac	d1	a2	a5	94	3d	45	a5	14	8e	d9	41	4b	87	8f	4e	=E....AK..N		
53	00000480	32	65	1b	08	f1	88	78	86	e3	1e	e0	3a	81	4b	1b	27	2e....x.....:K.'		
54	00000490	53	72	31	cc	af	23	89	a8	b1	29	cb	01	fc	81	56	1b	Sr1..#....)....V.		
55	000004a0	c8	9a	f2	95	3c	3c	82	a8	7a	1c	7f	8f	60	e7	f7	82	<..z....`...		
56	000004b0	66	1c	44	45	35	5c	47	a4	47	03	04	64	01	e8	b5	c4	f.DE5\G.G..d....		
57	000004c0	d0	40	0f	10	e7	d5	e5	e4	70	a5	24	bb	75	4d	6e	3a	.@.....p.\$..uMn:		
58	000004d0	33	b7	83	99	95	25	75	6a	56	15	aa	d0	be	e3	2d	7c	3....%ujV.....-		
59	000004e0	e9	c0	52	93	c9	96	e5	6d	1a	ff	24	16	87	8e	a7	68	..R....m..\$....h		
60	000004f0	6d	a4	6d	3f	07	a2	6c	72	3d	2d	1d	37	a7	9d	55	3c	m.m?..lr=-.7..U<		
61	00000500	c6	34	12	2e	58	25	73	64	1d	96	19	65	19	81	7d	d5	.4..X%sd...e..}.		