

Files and Filesystems



Things that computers do

1. Process information

2. Store information

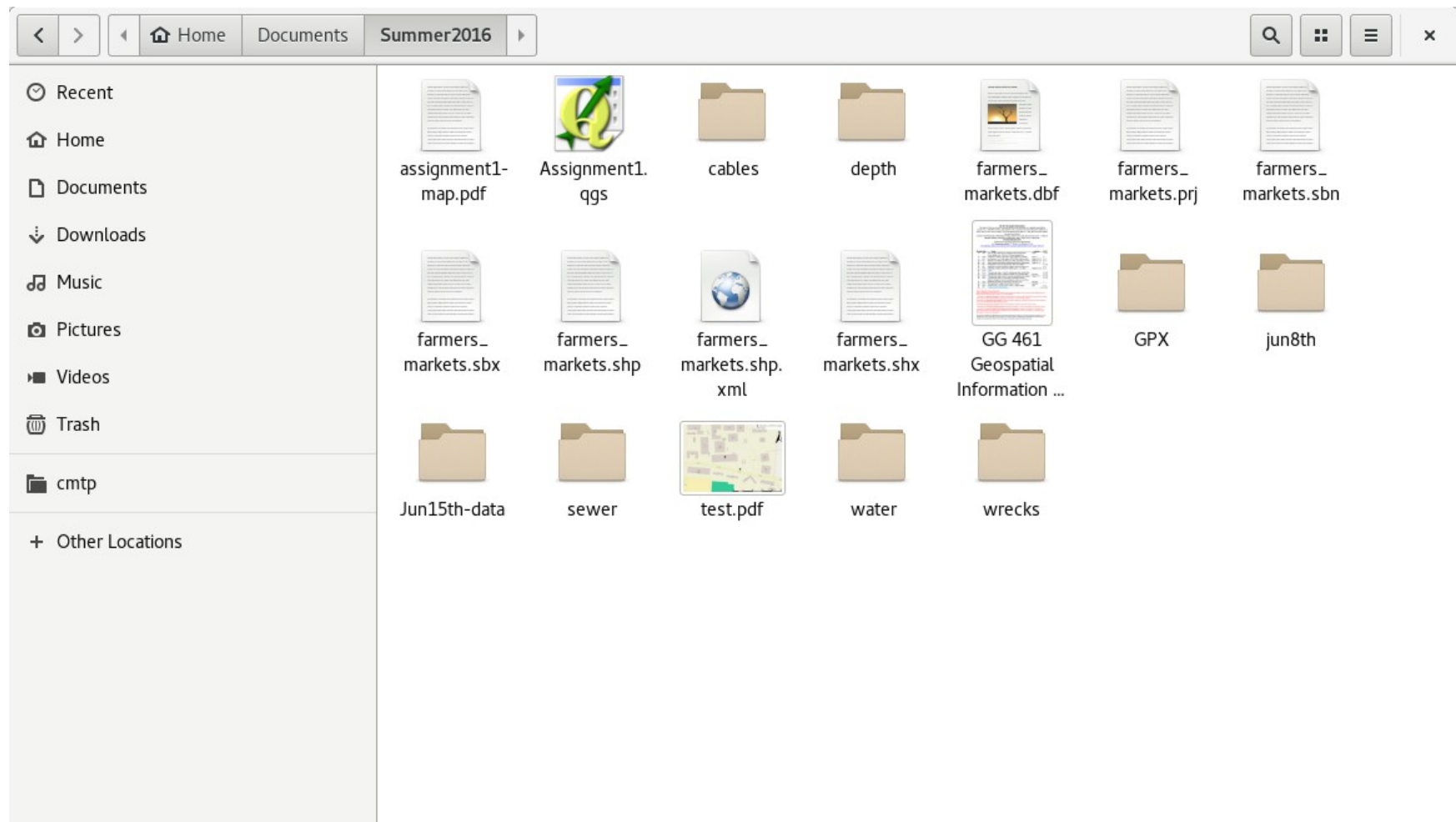


Things that computers do

1. Process information

2. Store information

What is a file?



What is a file?

- **A stream of bits**

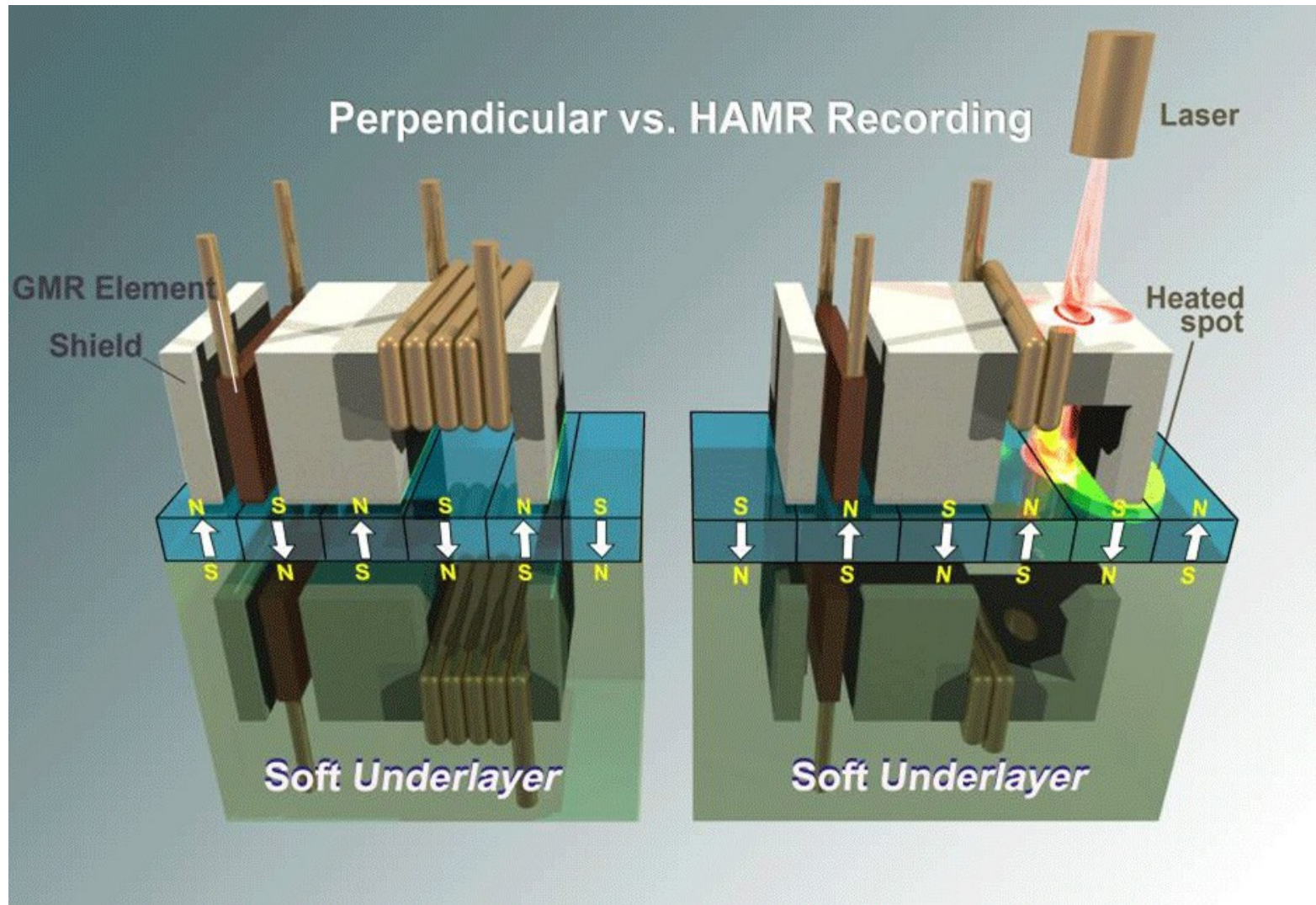
- 01110100001
- 101000001010101010000
- 11.....111111111

- **A stream of bytes**

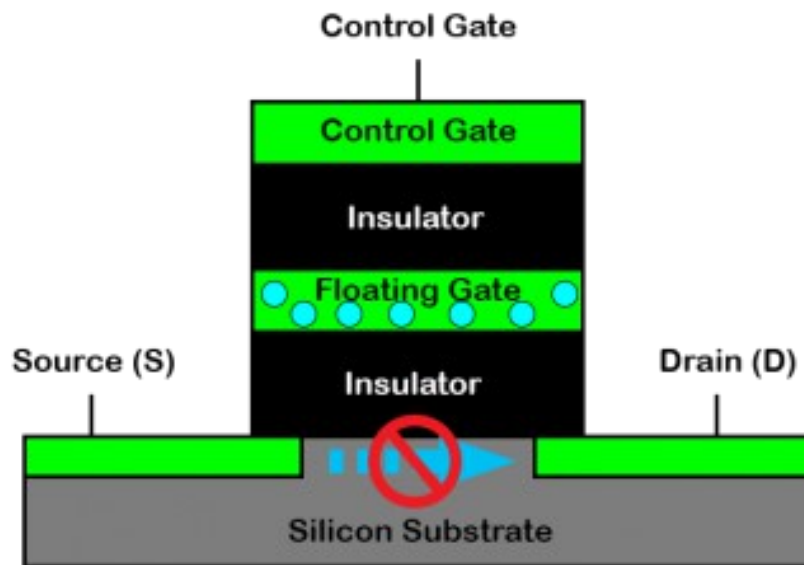
- [00110100]..[]..[11001101]



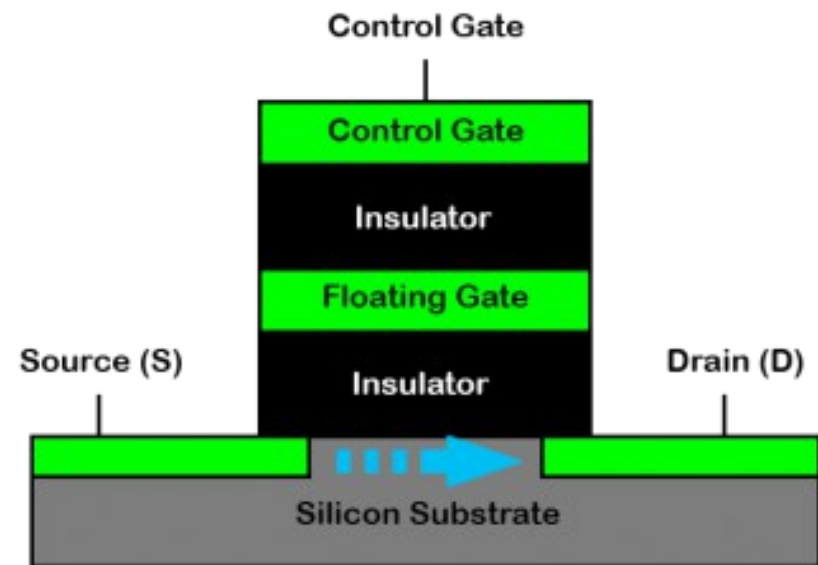
Physical Media



Physical Media

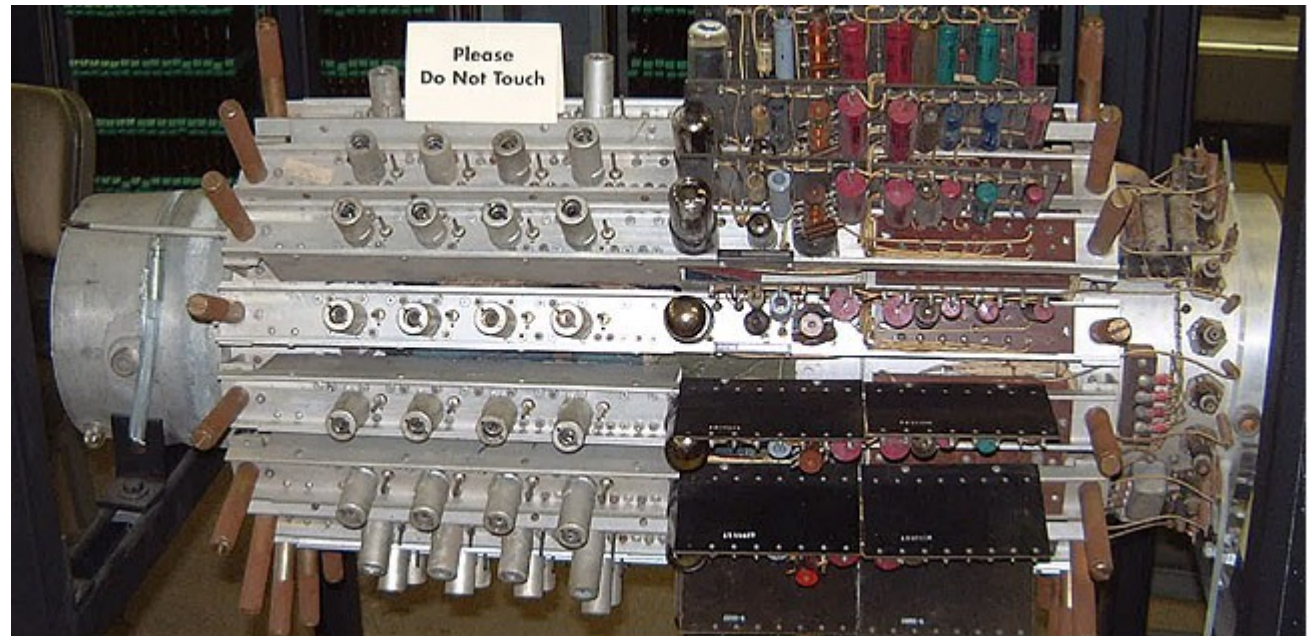
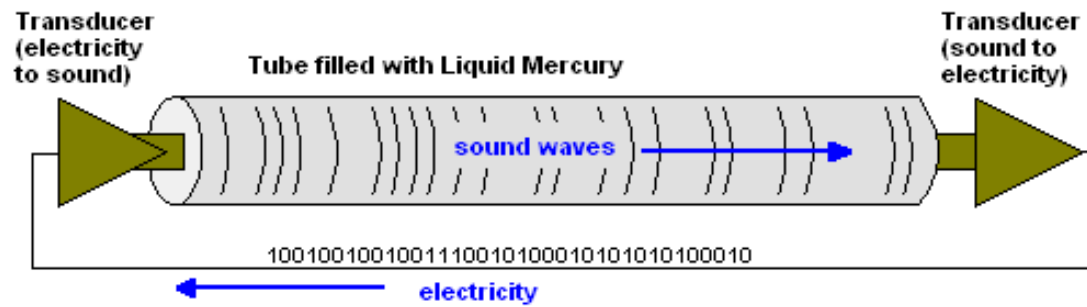


No Current - Floating Gate Programmed



Current Flows - Floating Gate Erased

Physical Media



Physical Media



Back to files

```
fish /home/jon/Documents/Presentations x
File Edit View Search Terminal Help
jon@localhost ~/D/Presentations> cat an-file
the quick brown fox jumps over the lazy dog
jon@localhost ~/D/Presentations> hexdump -C an-file
00000000  74 68 65 20 71 75 69 63  6b 20 62 72 6f 77 6e 20  |the quick brown |
00000010  66 6f 78 20 6a 75 6d 70  73 20 6f 76 65 72 20 74  |fox jumps over t|
00000020  68 65 20 6c 61 7a 79 20  64 6f 67 0a              |he lazy dog.|
0000002c
jon@localhost ~/D/Presentations> 
```

Dec	Hx	Oct	Html	Char	Dec	Hx	Oct	Html	Char	Dec	Hx	Oct	Html	Char
0	0	000		NUL	43	2B	053	+	+	86	56	126	V	V
1	1	001		SOH	44	2C	054	,	,	87	57	127	W	W
2	2	002		STX	45	2D	055	-	-	88	58	130	X	X
3	3	003		ETX	46	2E	056	.	.	89	59	131	Y	Y
4	4	004		EOT	47	2F	057	/	/	90	5A	132	Z	Z
5	5	005		ENQ	48	30	060	0	0	91	5B	133	[[
6	6	006		ACK	49	31	061	1	1	92	5C	134	\	\
7	7	007		BEL	50	32	062	2	2	93	5D	135]]
8	8	010		BS	51	33	063	3	3	94	5E	136	^	^
9	9	011		TAB	52	34	064	4	4	95	5F	137	_	_
10	A	012		LF	53	35	065	5	5	96	60	140	`	`
11	B	013		VT	54	36	066	6	6	97	61	141	a	a
12	C	014		FF	55	37	067	7	7	98	62	142	b	b
13	D	015		CR	56	38	070	8	8	99	63	143	c	c
14	E	016		SO	57	39	071	9	9	100	64	144	d	d
15	F	017		SI	58	3A	072	:	:	101	65	145	e	e
16	10	020		DLE	59	3B	073	;	;	102	66	146	f	f
17	11	021		DC1	60	3C	074	<	<	103	67	147	g	g
18	12	022		DC2	61	3D	075	=	=	104	68	150	h	h
19	13	023		DC3	62	3E	076	>	>	105	69	151	i	i
20	14	024		DC4	63	3F	077	?	?	106	6A	152	j	j
21	15	025		NAK	64	40	100	@	@	107	6B	153	k	k
22	16	026		SYN	65	41	101	A	A	108	6C	154	l	l
23	17	027		ETB	66	42	102	B	B	109	6D	155	m	m
24	18	030		CAN	67	43	103	C	C	110	6E	156	n	n
25	19	031		EM	68	44	104	D	D	111	6F	157	o	o
26	1A	032		SUB	69	45	105	E	E	112	70	160	p	p
27	1B	033		ESC	70	46	106	F	F	113	71	161	q	q
28	1C	034		FS	71	47	107	G	G	114	72	162	r	r
29	1D	035		GS	72	48	110	H	H	115	73	163	s	s
30	1E	036		RS	73	49	111	I	I	116	74	164	t	t
31	1F	037		US	74	4A	112	J	J	117	75	165	u	u
32	20	040	 	Space	75	4B	113	K	K	118	76	166	v	v
33	21	041	!	!	76	4C	114	L	L	119	77	167	w	w
34	22	042	"	"	77	4D	115	M	M	120	78	170	x	x
35	23	043	#	#	78	4E	116	N	N	121	79	171	y	y
36	24	044	$	\$	79	4F	117	O	O	122	7A	172	z	z
37	25	045	%	%	80	50	120	P	P	123	7B	173	{	{
38	26	046	&	&	81	51	121	Q	Q	124	7C	174	|	
39	27	047	'	'	82	52	122	R	R	125	7D	175	}	}
40	28	050	((83	53	123	S	S	126	7E	176	~	~
41	29	051))	84	54	124	T	T	127	7F	177		DEL
42	2A	052	*	*	85	55	125	U	U					

An ASCII file

An ASCII file (or text file as some call it) is a file consisting of byte which have values [0,127].

That is

00000000 (0x00)

To

01111111 (0x7F)

We use hex because it's more compact on screen and reading

000110101000101010110000101010101010101010101
0011110101

An text file

ASCII (American Standard Code for Information Interchange)

- **American is not the only language out there**
- **ASCII is only one encoding standard**
- **UTF-8 is more or less the new “text file” standard**

An binary file

Any file is binary by definition

Please don't use this terminology as it is deeply confusing



A note of file names



File Coding

- **ASCII - American Standard Code for Information Interchange**
- **File formats are agreed upon coding schemes**
- **I agree to read your bytes and ascii or utf or whatever**
- **You give me a hint as to how to read with a file extension**
- **Headers and footers help a lot, but add overhead**

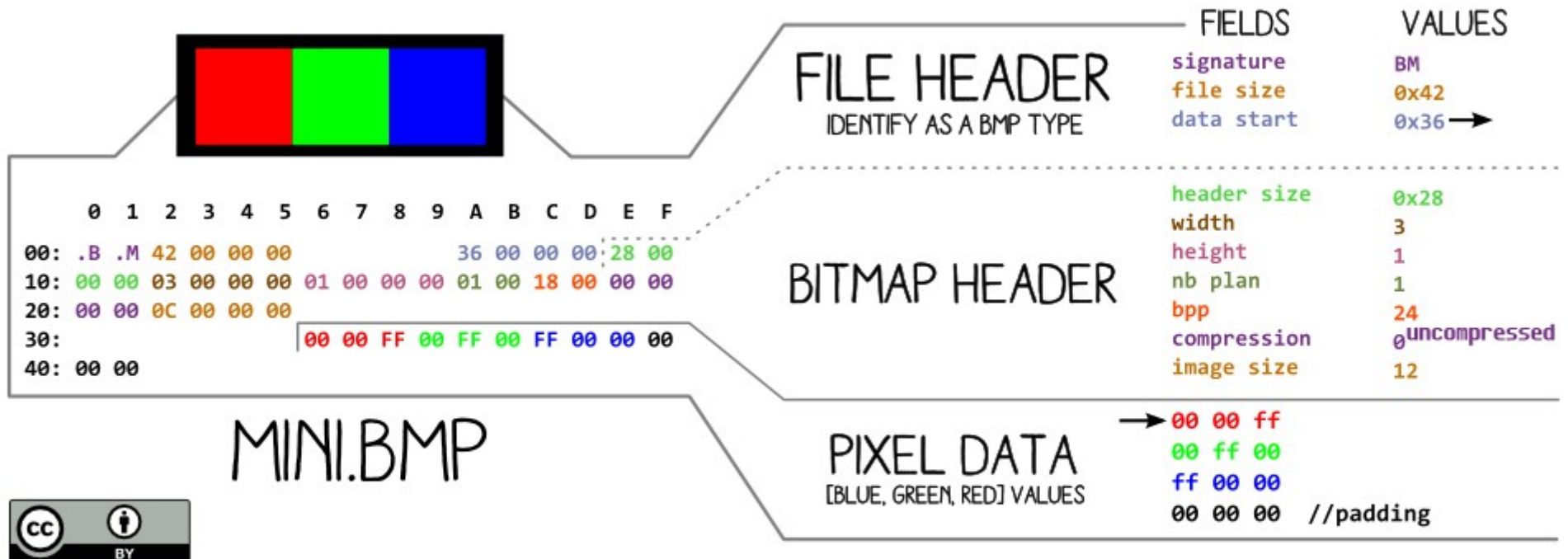
- **nothing stops me from reading how I choose**

Other coding standards

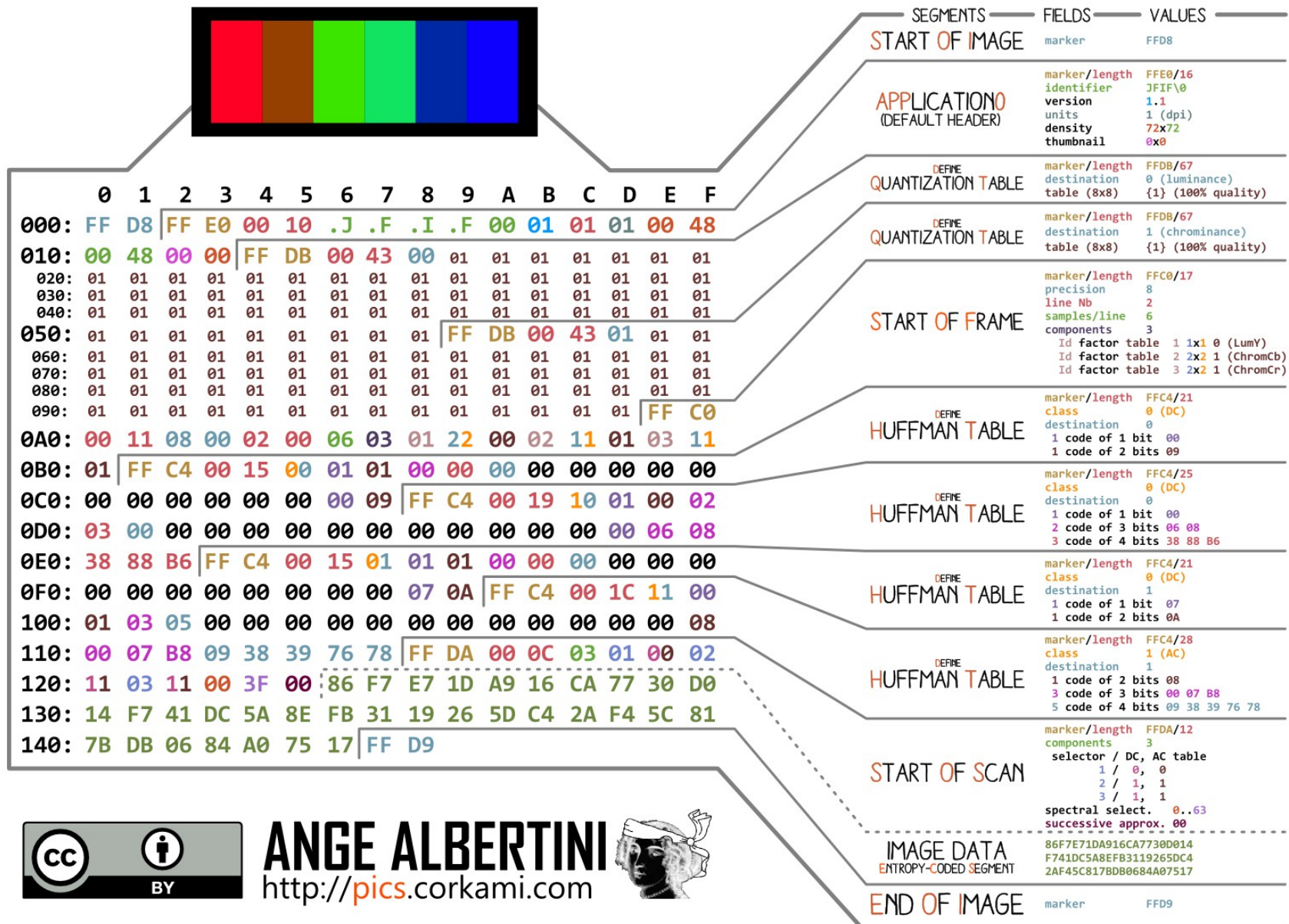
BITMAP / DEVICE INDEPENDENT BITMAP

SUBTYPE TYPE

ANGE ALBERTINI
<http://www.corkami.com>



JPEG FILE INTERCHANGE FORMAT



ANGE ALBERTINI
http://pics.corkami.com



JPEG IS THE ENCODING STANDARD, JFIF IS THE FILE FORMAT

Other coding standards

EXECUTABLE AND LINKABLE FORMAT

ANGE ALBERTINI
<http://www.corkami.com>



```
me@nux:~$ ./mini
me@nux:~$ echo $?
42
```

```

 0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00: 7F .E .L .F 01 01 01
10: 02 00 03 00 01 00 00 00 60 00 00 08 40 00 00 00
20:
    34 00 20 00 01 00
40: 01 00 00 00 00 00 00 00 00 00 00 00 08 00 00 00 08
50: 70 00 00 00 70 00 00 00 05 00 00 00
60: BB 2A 00 00 00 B8 01 00 00 00 CD 80
```

MINI

ELF HEADER

IDENTIFY AS AN ELF TYPE
SPECIFY THE ARCHITECTURE

FIELDS	VALUES
e_ident	
EI_MAG	0x7F, "ELF"
EI_CLASS, EI_DATA	1ELFCLASS32, 1ELFDATA2LSB
EI_VERSION	1EV_CURRENT
e_type	2ET_EXEC
e_machine	3EM_386
e_version	1EV_CURRENT
e_entry	0x8000060
e_phoff	0x0000040
e_ehsize	0x0034
e_phentsize	0x0020
e_phnum	0001
p_type	1PT_LOAD
p_offset	0
p_vaddr	0x8000000
p_paddr	0x8000000
p_filesz	0x0000070
p_memsz	0x0000070
p_flags	5PF_R PF_X

PROGRAM HEADER TABLE

EXECUTION INFORMATION

CODE

X86 ASSEMBLY

```
mov ebx, 42
mov eax, SC_EXIT1
int 80h
```

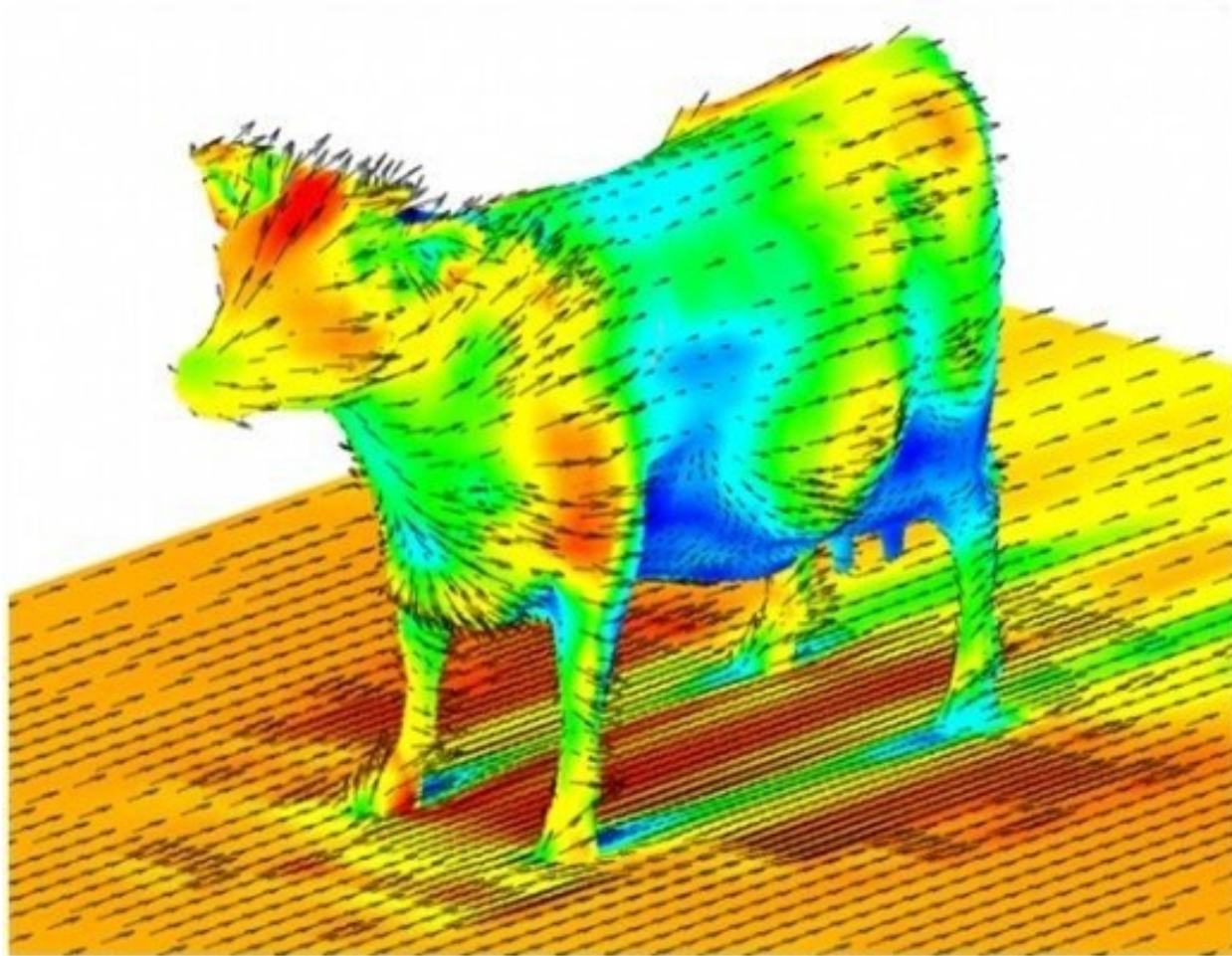
EQUIVALENT C CODE

```
return 42;
```

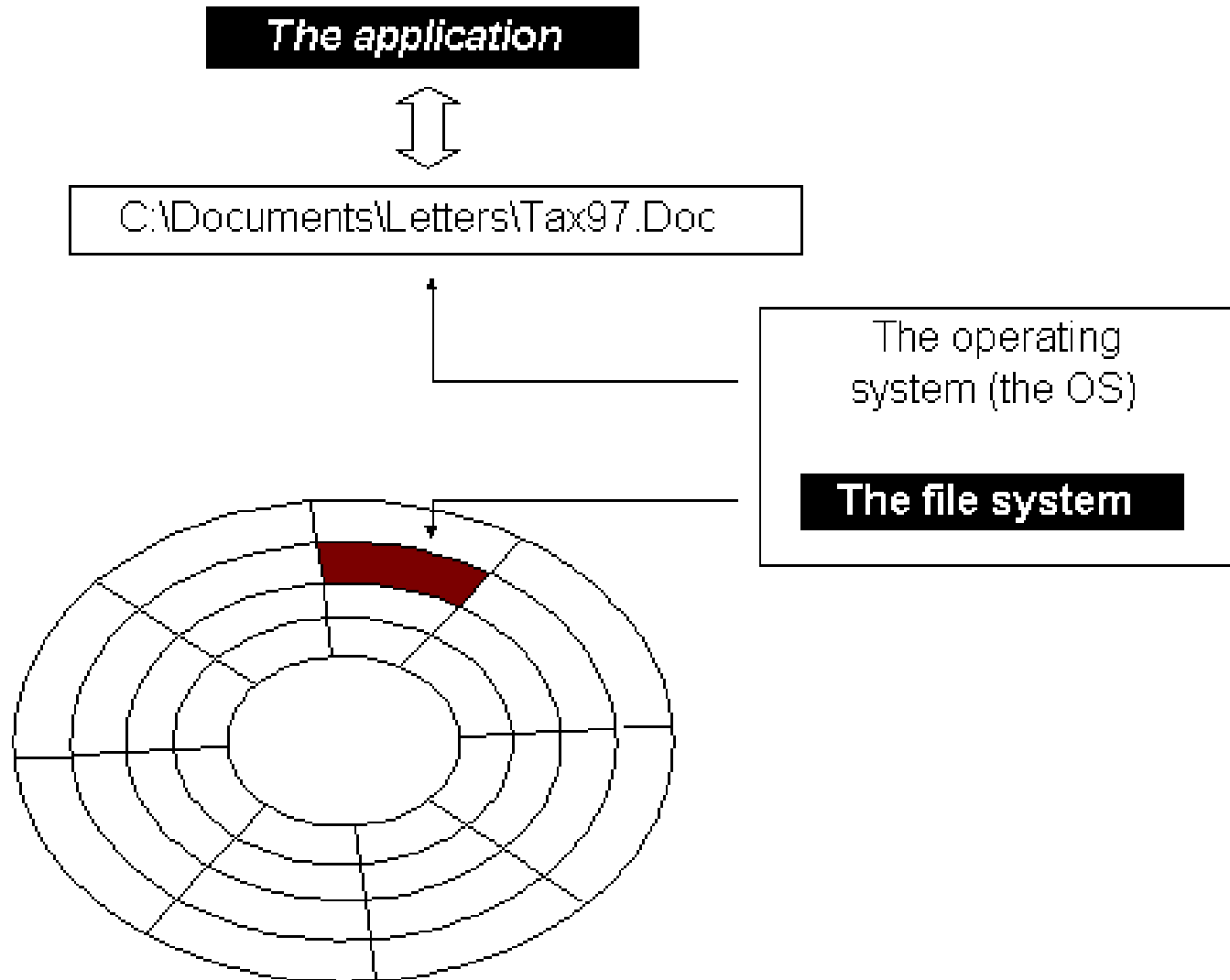
So files...

- **Just a bunch of ones and zeros**
- **You impose a meaning on it**
- **Files can be polymorphic (one data, many meanings)**
- **Can be nullmophic (is that a word? I'm gonna say that's a word)**
- **Live in in a larger stream of ones and zeros that is the storage device, which makes finding them fun**

Storing files

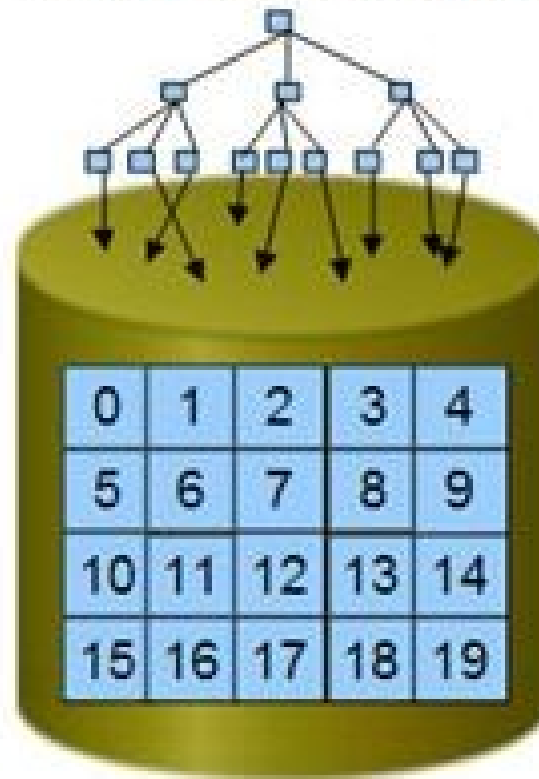


Storing files



Storing files

File names / inodes



Traditional
Hierarchical

Block Storage

- **Disks present themselves as Block Devices**

- There's a bunch of blocks/sectors
- Each block has a specific size (usually 4KB)
- Interfacing is done on a per block basis
- ie. "Write this to block #4" or "Give me the contents of block #4"



Block Storage

- **Turns out block storage is really inconvenient**
- **There file usage issues. How many blocks do I need for my file?**
- **Fragmentation issues. Is this block free for me to use?**
- **File systems do all this for us and give us a name space**



What is a file system

- **A file system is a way of abstracting away the physical realities of a storage medium**
- **It tracks blocks to name space objects (files)**
- **It takes care of all the pesky fragmentation, degradation, and performance issues**
- **In short it takes a chunk of raw storage and makes it a flexible storage medium.**

Filesystems

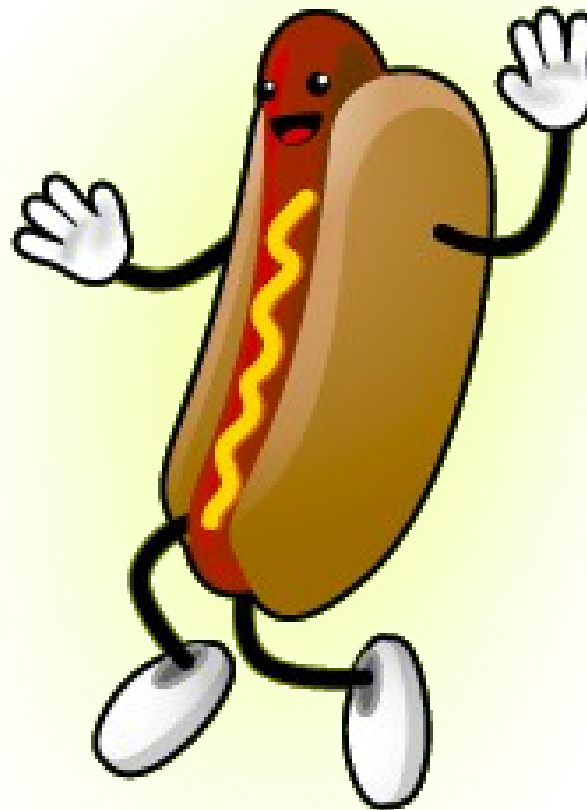
- **Range in size and designs**
- **Small ones (jffs), big ones (zfs/xfs), old ones (HFS), new ones (bcachefs), red ones (NILFS), blue ones (F2FS)**
- **Designed for ease of programming (databases)**
- **Designed for specific media. jffs/f2fs have ‘wandering logs’**
- **Designed for stupid scale.**
 - “If we could implement a physical system with the storage capacity that matches the 128-bit address range of ZFS, that we would literally evaporate all the oceans on earth “

https://blogs.oracle.com/dcb/entry/zfs_boils_the_ocean_consumes

Filesystems

- **The FAT family has no concept of users or permissions**
- **ZFS and btrfs do file integrity checking**
- **GlusterFS takes many and unifies them over network**
- **Many file systems have journals**
- **Etc... etc... etc...**
- **Different design goals lead to different file systems**
- **None are best, all are interesting**

Demo time



So file systems are files

File systems hold files, so file systems can hold file systems which can hold files (some of which could be file systems etc).

Further, files are just streams of ones and zeros across what could be random blocks of a storage device.

Hopefully everything is clear.



File carving

- **What do you do if you have a stream of ones and zeros and want something useful out of it?**
- **Failed storage device, filesystem failure, unknown filesystem, stupid programmer, etc...**
- **Tl;dr it's called file carving and it's exactly what it sounds like.**
- **https://en.wikipedia.org/wiki/File_carving**

Questions?

