

# COMP 3430

Operating Systems

June 26<sup>th</sup>, 2019

By the end of today's lecture, you should be able to:

- List types of file systems (in-class)
- Describe how a specific file system is implemented (e.g., FAT32) (FAT32 whitepaper, in-class)
- Evaluate the data structures used by a file system implementation (FAT32 whitepaper, in-class)
- Criticize a file system implementation (FAT32 whitepaper, in-class)
- Implement a read-only file system (FAT32 whitepaper)



© William Whyte, CC BY-SA 2.0

# What kind of RAID do I want?

Let's look at some scenarios and decide what RAID level would be appropriate for the problem, situation, or person.



Pixabay License

# Scenario 2: the gamer

RAID 0 or RAID 1?

This person wants **speed** .

Some things we need to consider:

- Type of workload (primarily random or sequential?)
- Bandwidth available from RAID level for that workload.

# RAID summary

Main takeaways:

1. A *hardware* (...or software) solution that balances reliability, capacity, and performance.
2. Implementation is *beneath* the file system.
  - The file system doesn't know anything about the *physical structure* of the drives.



Public Domain

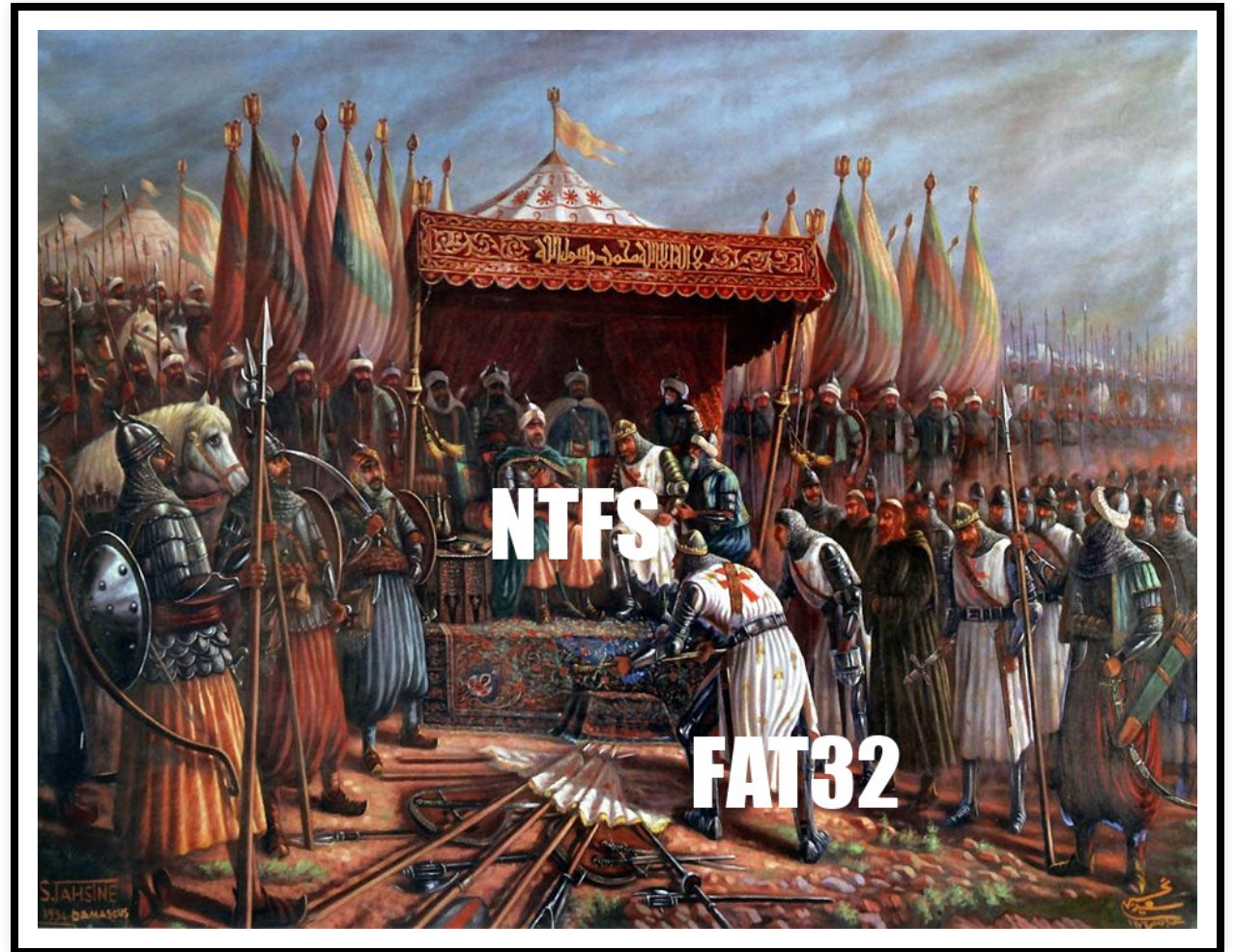
# File systems

- Many different kinds of file systems.

```
Disk /dev/nvme0n1: 477 GiB, 512110190592 bytes, 1000215216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 90400728-DEEA-4CB7-B7AD-8D1CDDC26D2E

Device          Start      End        Sectors    Size Type
/dev/nvme0n1p1    2048      1050623    1048576    512M EFI System
/dev/nvme0n1p2  1050624    2549759    1499136    732M Linux filesystem
/dev/nvme0n1p3  2549760  1000214527 997664768 475.7G Linux filesystem
```

/dev/nvme0n1 on my system.



Public Domain

# File systems

- Many different kinds of file systems.
  - `ext{2,3,4}`
  - `FAT{12,16,32}`
  - NTFS
  - APFS
  - ZFS
  - Btrfs
- **Think about it** : *Why* might there be so many different kinds of file systems?

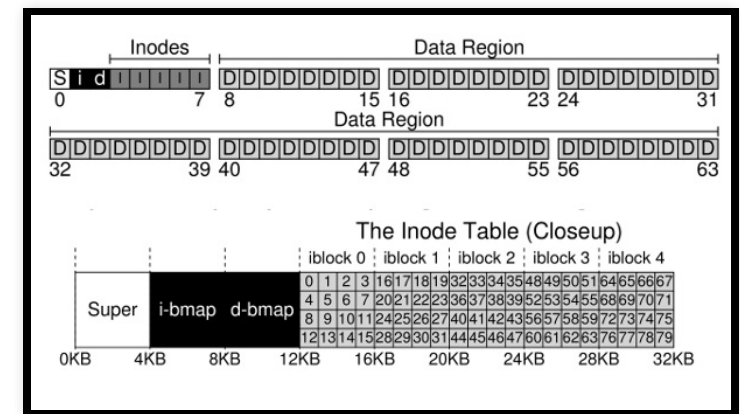




Robot is © Cory Doctorow CC BY-SA 2.0

# vsfs

- The textbook describes “vsfs”.
  - This is *remarkably* similar to ext2/3.
- Let’s remind ourselves about this file system.



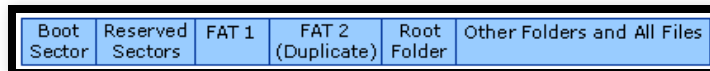
© 2008-19, Arpaci-Dusseau

# *Evaluating* the file system

1. Flexibility/expandability (can this FS handle gigabyte size files and terabyte size disks?)
2. Wasted space (how many bits are *wasted* by the FS?)
  - What *is* waste ?
3. Efficiency (how *fast* can I access a file with this FS? Can this FS become *fragmented*?)
4. Ease of implementation (can *I* write code to work with this FS?)
5. Data structures (how does this FS structure data on disk?)

# FAT32

- File Allocation Table comes in three variants: 12, 16, 32.
- Implementation is *somewhat* similar to vsfs & ext2/3
- Let's draw out what FAT32 looks like.
  - While drawing, we'll consider our evaluation questions.



© Microsoft Corp

00009800	43 4f 52 54 4f 31 20 20 54 58 54 20 00 64 55 b6	CORT01.TXT	8U
00009810	81 44 81 44 00 00 55 b6 81 44 00 00 19 00 00 00	D.D.U.D	
00009820	43 6f 00 2e 00 64 00 61 00 74 00 8f 00 b3 00 00	Co...d.a.t.	
00009830	ffffffffffffffffffffffff0000ffffffff		
00009840	02 6e 00 20 00 6e 00 6f 00 6d 00 0f 00 b3 62 00	n..n.o.m.	b.
00009850	72 00 65 00 20 00 6c 00 61 00 00 00 72 00 67 00	r.e..l.a..r.g.	
00009860	01 55 00 6e 00 20 00 61 00 72 00 8f 00 b3 63 00	U.n..a.r..c.	
00009870	68 00 69 00 76 00 6f 00 20 00 00 00 63 00 6f 00	h.i.v.o..c.o.	
00009880	55 4e 41 52 43 48 7e 31 44 41 54 20 00 00 59 b6	UNARCH-1DAT	Y
00009890	81 44 81 44 00 00 59 b6 81 44 00 00 20 00 00 00	D.D.Y..D	
000098a0	42 6f 00 00 00 ffffffff0000ffff	Bo.....Y	
000098b0	ffffffffffffffffffffffff0000ffff		
000098c0	01 74 00 65 00 72 00 63 00 65 00 8f 00 59 72 00	t.e.r.c.e..Yr.	
000098d0	5f 00 61 00 72 00 63 00 68 00 00 00 69 00 76 00	.a.r.c.h..i.v.	
000098e0	54 45 52 43 45 52 7e 31 20 20 20 20 00 00 59 b6	TERCER-1	X
000098f0	81 44 81 44 00 00 59 b6 81 44 00 00 18 00 00	D.D.X..D	

DOS filename	DOS attributes
Reserved	Creation date and time
Last access date	Modification date and time
High bits of the first FAT32 cluster	Low bits of the first FAT32 cluster
File size in bytes	VFAT extended name
Extra attributes (always 0x0f)	Type (always 0x00)
Checksum of DOS name	First DOS cluster (always 0x0000)
Extra space after the extended filename	Sequence number for the extended filename entry

© Gunnar Wolf (CC BY-SA 3.0)

# Think about it ⌚

After evaluating the two file systems, can we  
*objectively* say that one file system  
implementation is better than another?



Pixabay License

You should be able to:

- List types of file systems (in-class)
- Describe how a specific file system is implemented (e.g., FAT32) (FAT32 whitepaper, in-class)
- Evaluate the data structures used by a file system implementation (FAT32 whitepaper, in-class)
- Criticize a file system implementation (FAT32 whitepaper, in-class)
- Implement a read-only file system (FAT32 whitepaper)



© William Whyte, CC BY-SA 2.0



It's only physics if it's from the  
Physique region of France.  
Otherwise it's just sparkling math

18:54 · 22 Jun 19 · [Twitter Web App](#)

**57** Retweets **194** Likes

