COMP 3430

Operating Systems

June 26th, 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)

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.

Scenario 2: the gamer

RAID 0 or RAID 1?

This person wants **speed** .

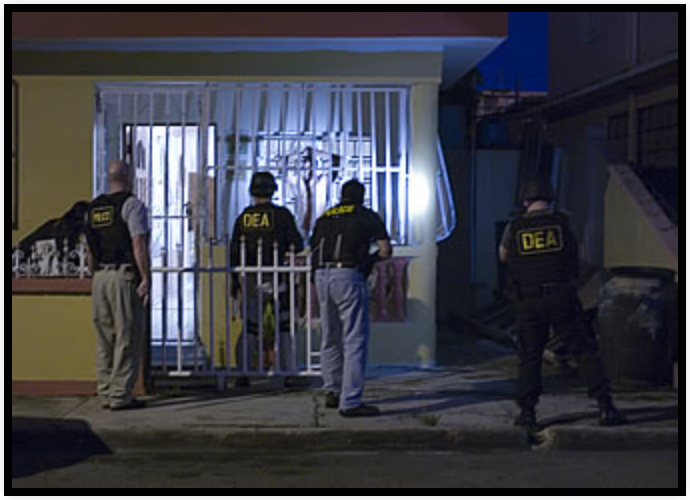
Some things we need to consider:

- 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

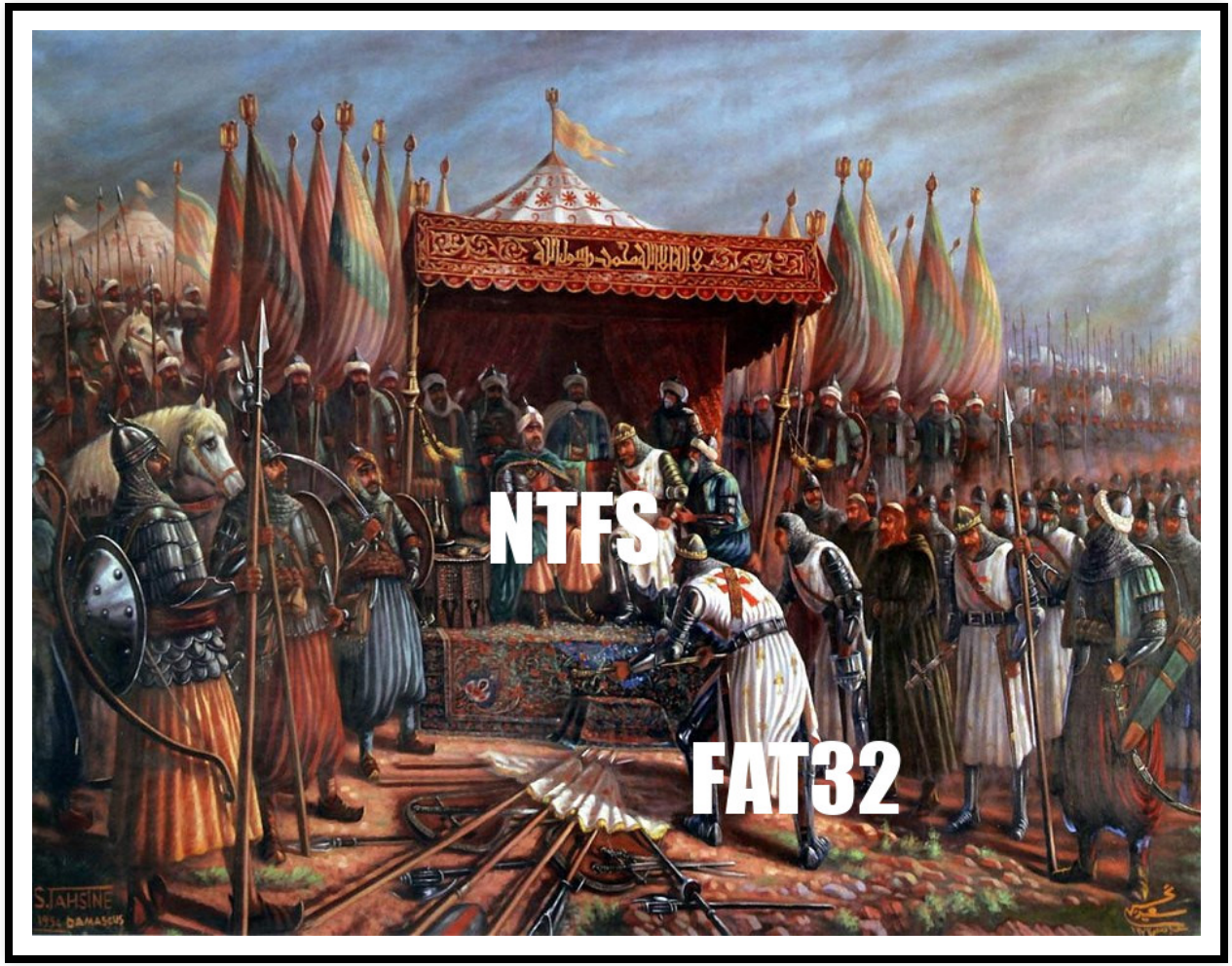
File systems

- 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

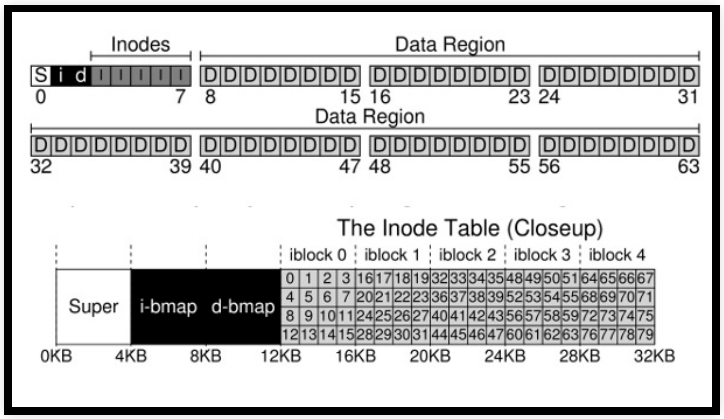
File systems

- 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.

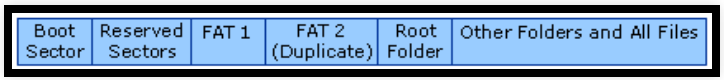


Evaluating the file system

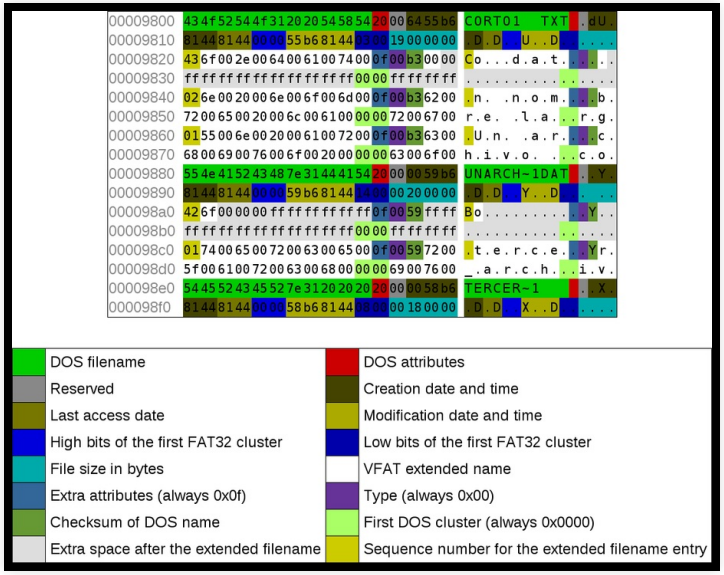
1. Flexibility/expandability (can this FS handle gigabyte size files and terabyte size disks?)
2. Wasted space (how many bits are *wasted* by the FS?)
   - What *is* waste ?
3. Efficiency (how *fast* can I access a file with this FS? Can this FS become *fragmented*?)
4. Ease of implementation (can *I*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

FAT32

- **F**ile **A**llocation **T**able 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



© Gunnar Wolf (CC BY-SA 3.0)

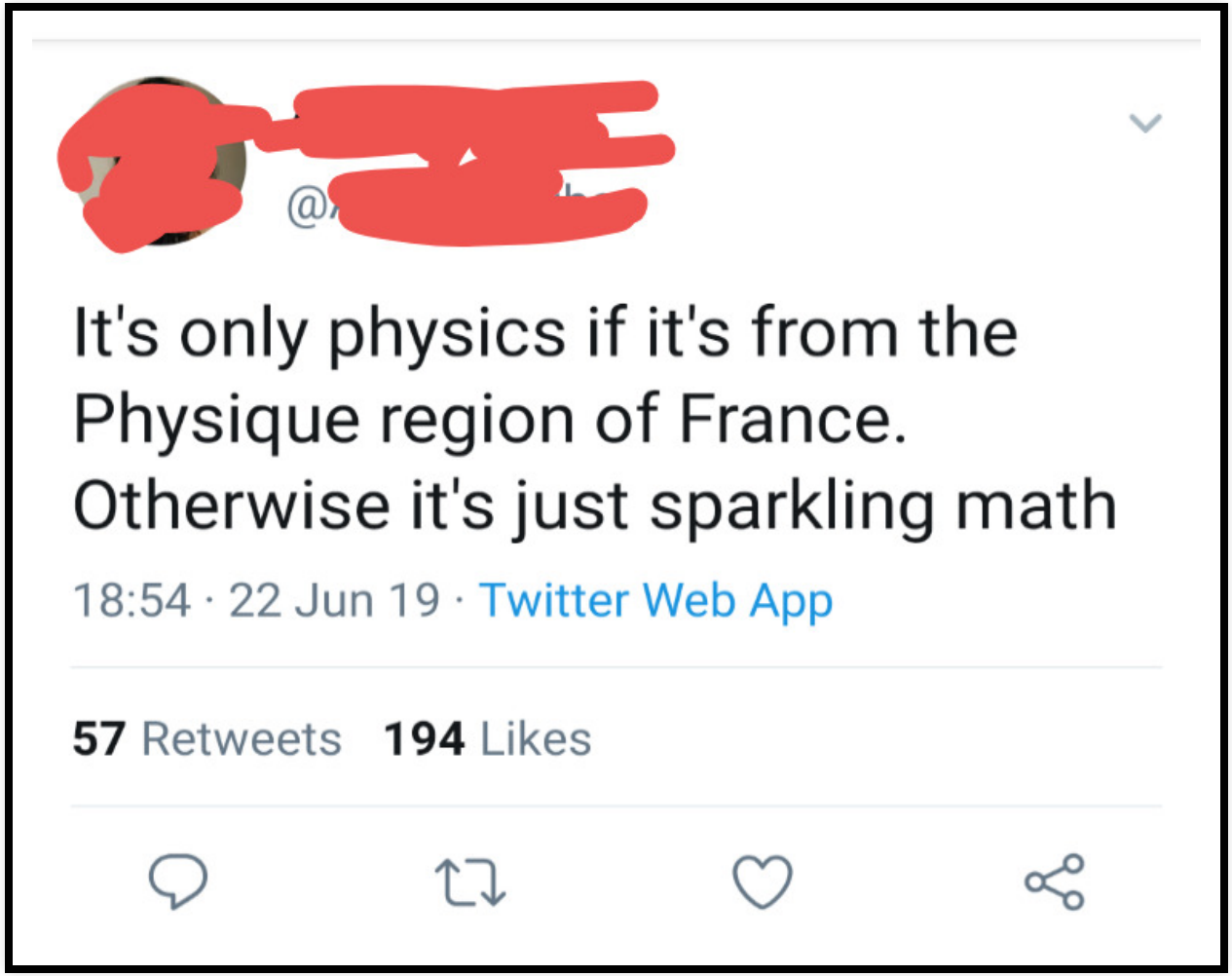
Think about it ⌛

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)


@
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