COMP 3430

Operating Systems

June 17th, 2019

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

- Explain how ~~a file~~ information is physically stored on disk. (chapter 37, in-class)
- Evaluate whether or not metadata is required by the OS. (chapter 39/40, in-class)
- Describe abstractions used to represent files and folders. (chapter 39)
- Use system calls to create, read, and write files and directories. (chapter 39)
- Describe the data structures that the OS uses to represent files. (chapter 39/40, in-class)
- Compare and contrast I/O scheduling and process scheduling. (chapter 37, in-class)

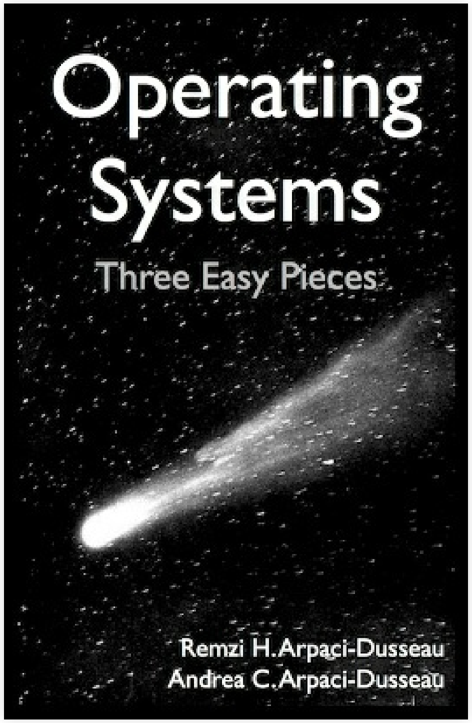
© Torsten Bolten CC BY-SA 3.0

Assignment 3

OSTEP Q ’n A

Chapters for ~~this week~~ today:

- Chapter 37
- Chapter 39
- Chapter 40



© RH & AC Arpaci-Dusseau

Hard drives

Let’s uh, look at a hard drive.

- Note: *modern* drives use 4k sectors

“1956 5MB IBM hard drive” License… unknown?

Data structures

Two halves to this discussion

1. How *process* data structures hold onto file information
2. How a *filesystem* jams bits into a drive.

© Franklin Bristow (2014)

inodes

- The book describes “vsfs” (but, importantly, doesn’t give you any code…)
    - The book *describes* the idea of direct pointers, singly indirect pointers, doubly indirect pointers, trebly indirect pointers.
        - … it doesn’t say anything about quadruply indirect pointers.
    - Leaves interpretation up to the reader.
- **Think about it**: How does the book’s implementation of an inode differ from the ext2 implementation of an inode?
    - Let’s *draw* an inode from ext2.

Metadata

- Filesystems have *a lot* of metadata.
  - **Think about it**: Let's agree on the definition of a word: what *is* metadata?

Do we need *that* metadata?

Think about it: Let’s look at some *real* metadata and decide if the metadata is necessary in contrived situations.

Situation #1: it’s just me

I am **literally** the only person using this machine and this OS. I picked a **super strong** password (`hunter2`).

Do I *really* need to have all of these ACL, permission, users, and groups metadata?

Pixabay License

Situation #2: “Deletion time”

I can’t even come up with a contrived situation for this. The ext2 filesystem has a field for “Deletion time”. What *possible* use could this have?

Pixabay License

Situation #3: `atime`

OK, I still can’t come up with a contrived situation.

¯\_(ツ)_/¯

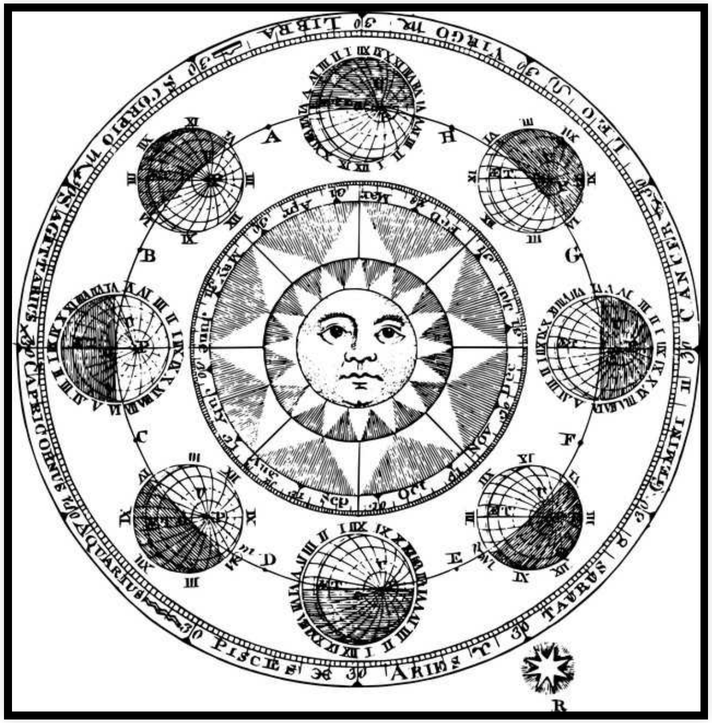
You come up with a situation: When might “last accessed time” be useful?

- *Note*: `atime` can be *disabled*.

© DEVON
(This costs $18,450 USD @ watches.com)

I/O Scheduling

- I/O scheduling and process scheduling are *remarkably* similar.
  - Time remaining ↔ # sectors/blocks to write
- … but there's *one* **fundamental** difference.
  - **Think about it**: What is *fundamentally* different between I/O scheduling and process scheduling? (hint: why did we *abandon* algorithms like shortest remaining time first?)



Pixabay License

Cool stuff: `strace`

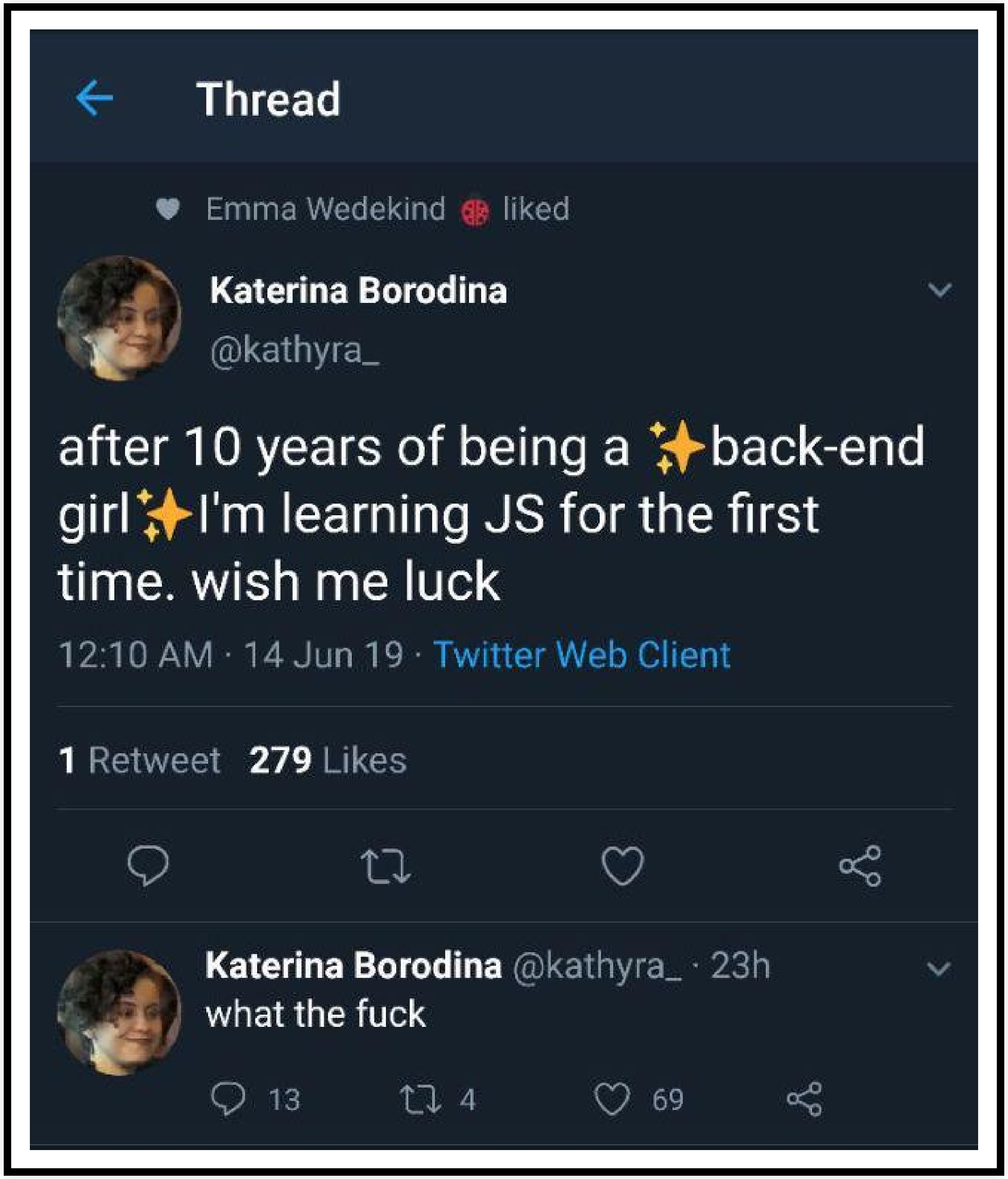
Let's look at `strace`.

Now you, too, can be a hacker. (Pixabay License)

Goals

You should be able to:

- Explain how ~~a file~~ information is physically stored on disk.
- Evaluate whether or not metadata is required by the OS.
- Describe abstractions used to represent files and folders.
- Use system calls to create, read, and write files and directories.
- Describe the data structures that the OS uses to represent files.
- Compare and contrast I/O scheduling and process scheduling.



Thread

Emma Wedekind liked

Katerina Borodina
@kathyra_

after 10 years of being a ✨back-end girl✨I'm learning JS for the first time. wish me luck

12:10 AM · 14 Jun 19 · Twitter Web Client

1 Retweet **279** Likes

Katerina Borodina @kathyra_ · 23h
what the fuck

13 4 69