

# Recitation 4: Unix I

MIT - 6.033

Spring 2022

Henry Corrigan-Gibbs

# Plan

- \* Unix history & context
- \* Discussion: files, fds
- \* Breakout rooms:
- \* Wrap-up discussion

## Logistics

- \* Feedback form  
feedback.henrycg.com
- \* Recitations finalized?
- \* Volunteers for rec questions.

# Unix History

Who are the authors?

Where are they?

Ritchie: C

Why are they qualified?

(6.033 recitation notes ... back to 1975)

Counterpoint to Multics — all-encompassing (Project MAC)

↳ Delays in development in Multics caused Bell to pull out

Computer system designed by programmer vs business person

Biz: More stuff → \$\$\$\$ (e.g. spreadsheet file, PDF file ...)

Prog: More stuff → More bugs & more headaches

"Small is good"  
↳ "Worse is better"

→ Show slides ←

↳ Relation of Unix to later systems

**IMPORTANT:** The system you use today is influenced by Unix, but is WAY more complicated than Unix.

# File I/O in Unix

- What is special / different about files in Unix versus predecessors?

\* Special files — everything is a file (mouse, elevator)

/dev/random  
/dev/null  
/dev/mem  
/dev/input/mouse  
⋮

Does any special file have a corresponding device?

- Simple interface — just R/W

↳ OS has no idea what's in your files.

- Why is this a good idea? Bad idea?

Is fd returned same as inode of file?

# Other ways it could have worked

## Memory - Mapped Files

```
ptr = mmap("my-file.txt");  
ptr[2] = 'a';
```

- + Simple
- + Intuitive
- Addr space is small
- Resizing
- Maybe not eff
- Performance?

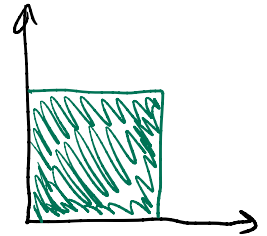
## Key - Value Store

```
ptr = read(filep, "key")  
write(filep, "key", "value")
```

- + Good for many apps
- Not good for all apps
- OS has to make choices... will probably choose wrong

## 2D Files

```
read(filep, buf, xmin, xmax,  
      ymin, ymax);
```



- + Writing 2D shapes
- + Fancy APIs
- Complex
- Not easy to map to devices
- App-specific



## Links

- What is a link? Symlink?

↳ Why is a hardlink useful?

\*Don't need to worry where "real file" is

## Set uid bit

Why can't anyone just set that bit?

## Disk blocks

Why 512 bytes?

## Discussion

2. In §3.2: "all links to a file have equal status"

↳ What does this mean?

↳ What are pros/cons to this approach?

3. What is an example of simplicity = generality?

\* File size limit

\* GUI



i-list

i-node

user ID, group ID  
access bits  
phy addresses (<10)  
size  
times  
# links  
type: dir/Sib/----

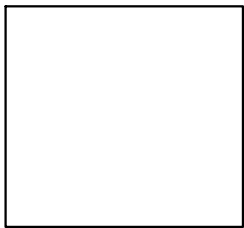
1. Find i-node for root dir
2. Open physical addr with contents
3. Look for /home
4. Find i-node # of /home in dir
5. Look at i-node of /home  
Open physical contents of /home

\* Find loc of ...

Lots of extra steps  
(perm checks, times, etc.)

# Big files

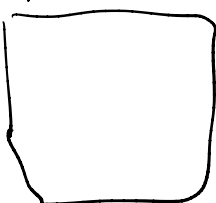
10 dir blocks



inode



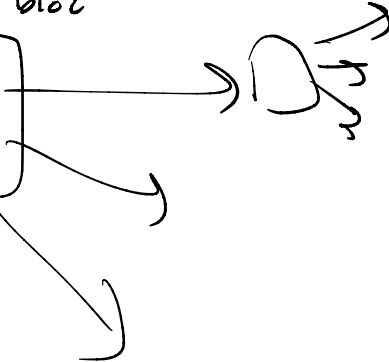
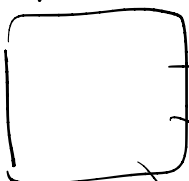
indir block



128 indir blocks



indir indir bloc



## Wrap-up Discussion

1975: Is the lack of objectives key to building successful systems?

1976: How might lack of objectives contribute to failure?

Consequences of "by programmers"?