Recitation 4: Unix I
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 $$ \rightarrow \$$ SpreadSheets

Prog: More stuff $$ \rightarrow \$$ More bugs & more headache

"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 (elevator)
    
    /dev/random
    /dev/null
    /dev/modem
    /dev/input/mouse

- Simple interface — just R/W
  Linux 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**

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

**Key-Value Store**

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

**2D Files**

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

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

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

+ Writing 2D shapes
+ Familiar APIs
- Complex
- Not easy to map to devices
- App-specific
Exercise - As a class, draw!

- How does fs know where to put new data on disk?
  - `mkdir("/home")`
  - `create("/home/file.txt")`
  - `create("/home/big.txt")`

- What happens if machine crashes midway?
  - `link("/home/big.txt", "other.txt")`

Here is the file:

```
"Start of big file"
```
Links

- What is a link? Sym link?

  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?
2. In §3.2: “all links to a file have equal status”

   a. What does this mean?
   b. What are pros/cons to this approach?

3. What is an example of simplicity = generality?
   * file size limit
   * GUI
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 ...

* Lists of extra step
(pem checks, times, etc.)
Big files

inode

10 dir blocks

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"? 