Intro

(Fall-18)

→ Why OS?

→ Background

→ OS : demo
   + roles (VM + lib + ...)

→ Course overview

→ History

→ Conclude

CS 537 Operating Systems

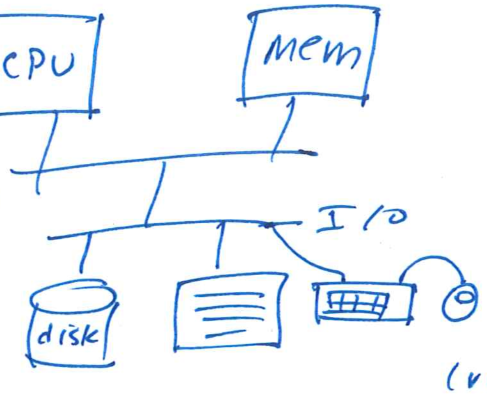
Why study OS?
- key to knowing "how stuff works"
- some conceptually interesting/beautiful stuff
- first steps towards real understanding (datacenters, etc)

Background needed

Basic programming : CS 367 (etc.) [354 => C]

Intro to systems : CS 354

=> how a computer basically works



2b not a big distance between C <-> machine

2 problem:

```
void func(int x){
    x = x+1;
}

int y=10;
func(y);
```

Q) what happens?

1 => discuss life cycle of a (running) program

=> load disk -> mem

=> execute ("run") [fetch, decode, execute]

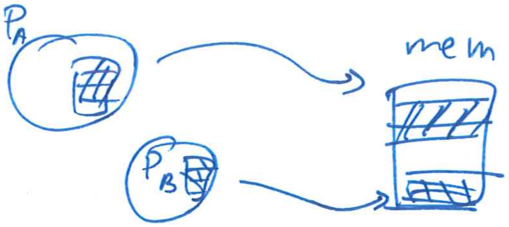
key: memory where code/data lives

2 => higher-level: life of C program

Focus thus far: mostly on one program

if that's all you need, done (no OS)

OS : intro via demo
- CPU: more than one prog @ time
- Mem: same
- I/O: show trace
- Concurrency: show weird behavior



OS roles

Virtual machine

1 CPU, 1 memory $\overset{\text{seems like}}{\Rightarrow}$ many CPUs, many large/private mems

CPU: switch between many progs (time share)

Mem: share in space (divide up)
(share)

Standard libraries

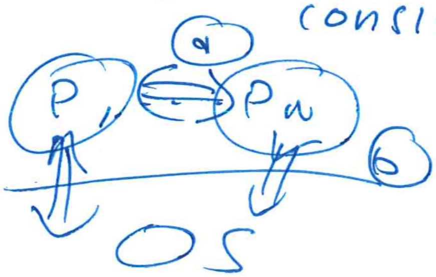
make system easy to use

(fopen() vs. I/O stack bit twiddling)

Do so while being:

efficient => can't do w/ excess overhead

secure => can't do w/o considering security



Course overview

[Web Page]

⇒ Overview → Materials (Book)
→ Mailing
→ Projects
→ Exams

⇒ Schedule
⇒ (connect to reading)

⇒ Videos

⇒ etc.

History

A little on organic development of OS

(mainframe) → (everything today)

principles learned over the years
⇒ now applied ubiquitously

Conclude

what is a class? → me
→ but also you

(Story of the OS book)

Fin.