

# CS 202-(001): Operating Systems

Instructor: Michael Walfish

TAs: Zach DeStefano (head TA)

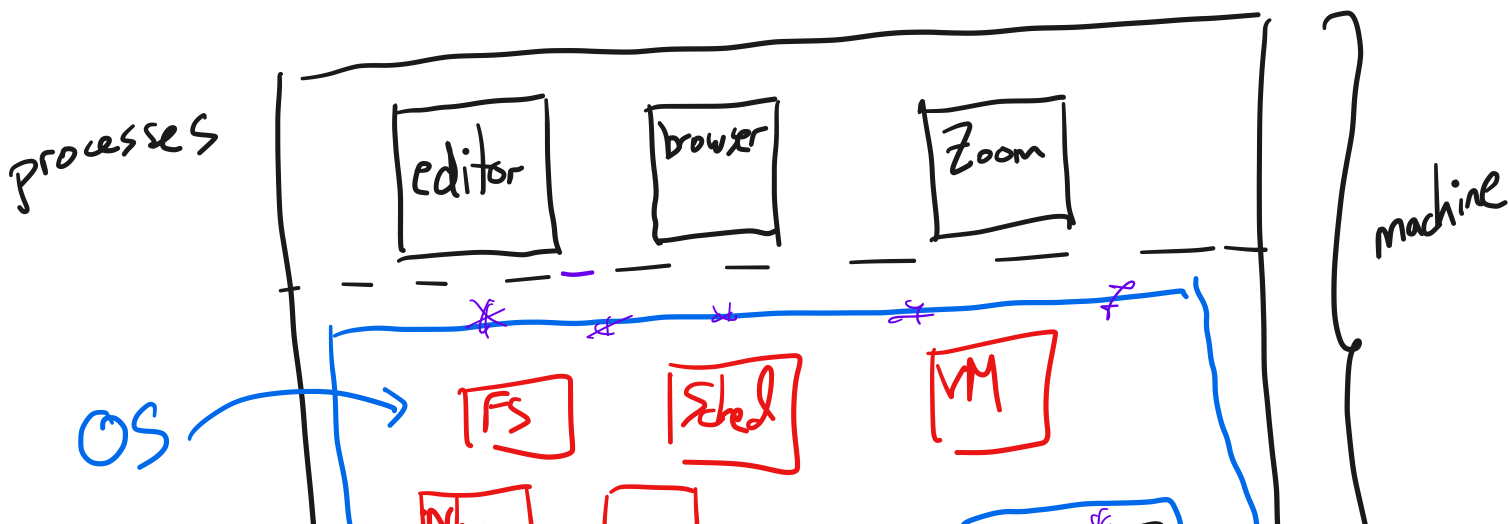
Charlie Chen      Jinli Xiao

Morgan Xu      Richard Xu

<http://www.cs.nyu.edu/~mwalfish/classes/23sp>

- 1. Intro & goals
- 2. What is an operating system?
- 3. Why study systems?
- 4. How will we study (operating) systems?
- 5. Mechanics + admin
- 6. History
- 7. Processes (next time)

## 2. What is an operating system?





Classical description of OS: isolation

- I. Managing the resources of the machine
- II. Abstracting the hardware

Examples?

- file system

```
int fd;
fd = open("/tmp/foo", O_WRONLY);
write(fd, 26, "abc...z");
```

df/folder

Abstraction: continuous array of bytes

Isolation: perms, "own view"

- Text input

Abstraction: whole (what keyboard)

Isolation: right application

- Memory

Abstraction:

```
movl 0x1248, %rax
```

Isolation: one process can't modify another's

- Scheduling
  - Abstraction :
  - Isolation :
- 

3. Why study systems?
- "how things work"
  - ideas are everywhere
  - fundamental design trade-offs
  - unsolved problems
  - skills building
- 

4. How will we study?

The Lions Book, John Lions

---

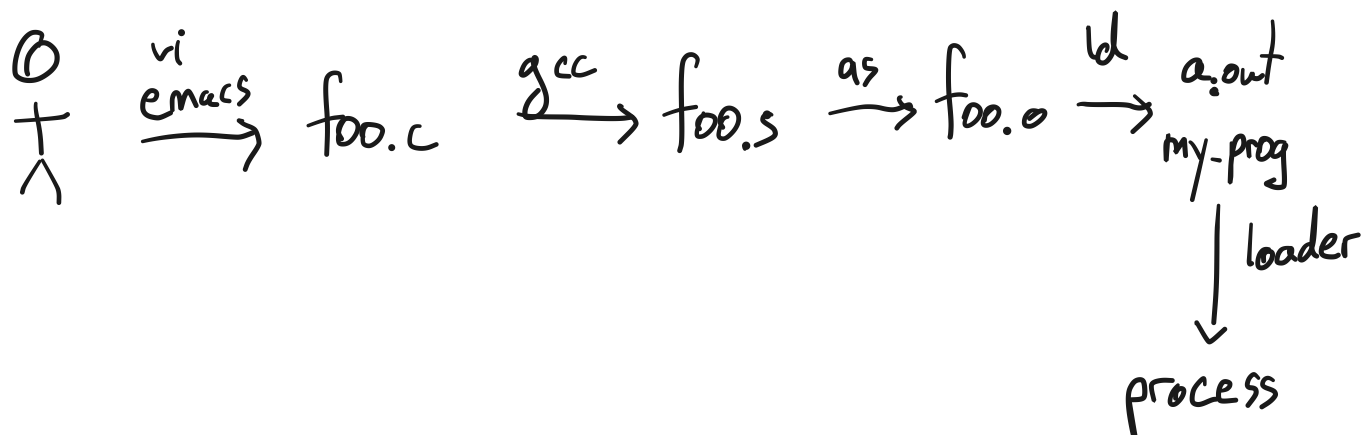
5. Mechanics + admin

- Comms
  - Components
    - class
    - labs
    - exams 3/8/23
    - reading
    - H/w (≠ labs)
    - recitation / review; some Ths 5pm - 6pm
  - grading
  - policies
- 5% HW  
 25% Labs  
 25% Midterm  
 45% Final

6. History (abridged)      Unix

7. Processes

Key abstraction

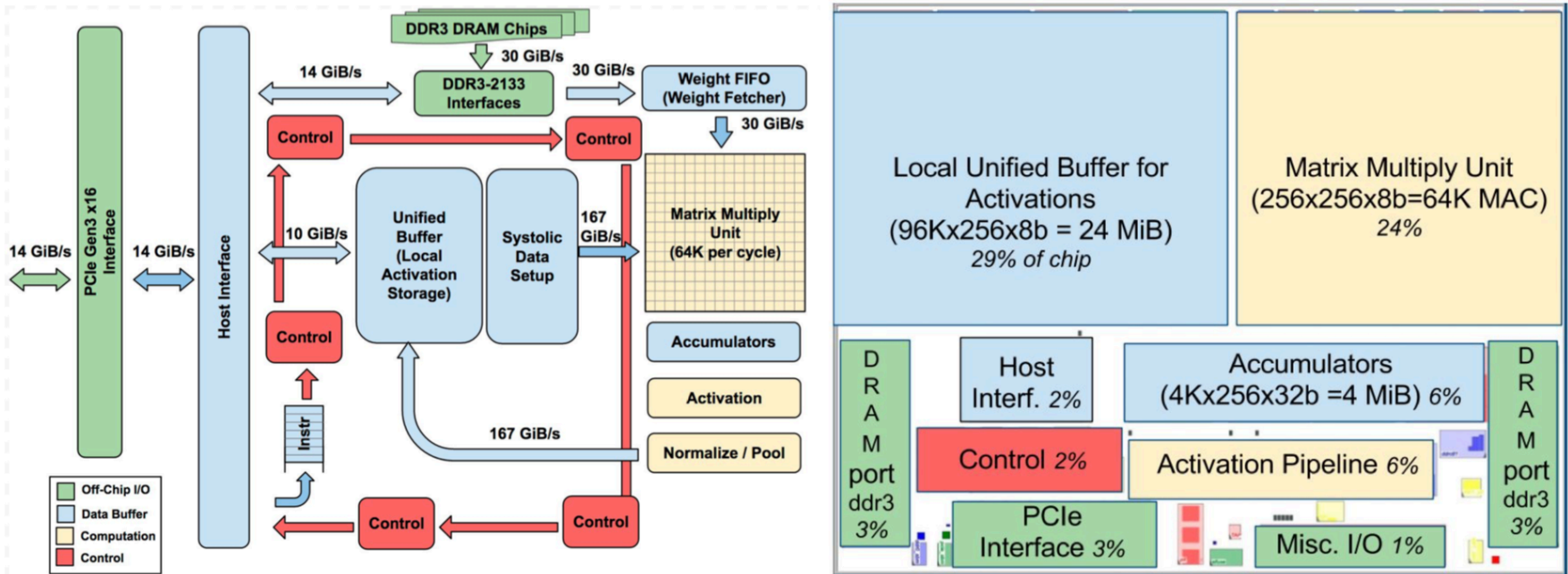


processes can be understood in two ways:

- from the process's point of view

- from the OS's point of view

# C? x86-64? My Future Is In Machine Learning!



- Cutting-edge ML backed by custom TPU, unique system software and OS support...



(see class notes for source)





(See class notes for source)