Welcome to 451!

OS: a program that abstracts & manages *hardware resources*.

for user program.

What are they?

In this class we care about CPU, DRAM, & storage devices.
CPU

CPU: Central Processing Unit

- Control Unit
- Arithmetic / Logic Unit
- Registers: PC, AC, MAR, MDR

Memory Unit

Output Device

Input Device

- executes instr.
- % rip holds address of next instr. to execute.
- OS sets up % rip for a new process.
Physical memory

- byte-addressable, slower than CPU
  (why we have caches)

Volatile: data does not last through a power cycle
Storage Devices

- persistent, large capacity (TBs) (non-volatile)
- block-addressable, way slower than memory

Performance also differs based on access patterns (sequential or random)

Hard drive
- sector size 512 bytes

Solid state drive (SSD)
- page size 4096 bytes
Other I/O Devices

Input: mouse, keyboard, webcam, microphone

Output: monitor, headphones, speakers
How does the OS abstract these resources?

CPU => process
DRAM => virtual memory
Storage => Filesys (files, directories)
Network => Network Stack (TCP/IP)
Why provide them?
- **Ease of use**
- **Simplified services**
  - Storage -> filesystem: named files, folder organization
  - DRAM -> virtual memory: a process owns the entire address space
- **Mask HW limitations**
  - Filesystem: file handles bytes (hides blocks)
  - Virtual memory: allows process to use more than physically available
- **Common interface**
  - Allows processes to share & communicate
  - Programs portable across hw
managed access [referee].

resource management

- schedule processes onto a single CPU (saves & restore each process's state)
- sharing of physical memory

- isolation (or OS)
  - a process can't read other processes' memory

managed sharing
  - explicitly requested shared memory

How does OS provide these abstractions?

What this class is about!