
# CSCI 3753 Operating Systems
# Summer 2019

**Christopher Godley**

**PhD Student**

**Department of Computer Science**

**University of Colorado Boulder**

# Welcome to Operating Systems

- Time: M/Tu/Th/F 9:15 am – 10:35 am
- Location: KTCH 1B84
- Office Hours: Tu/Th 12:45 am – 1:45 am
- Instructor: Christopher Godley

- CAs: TBA

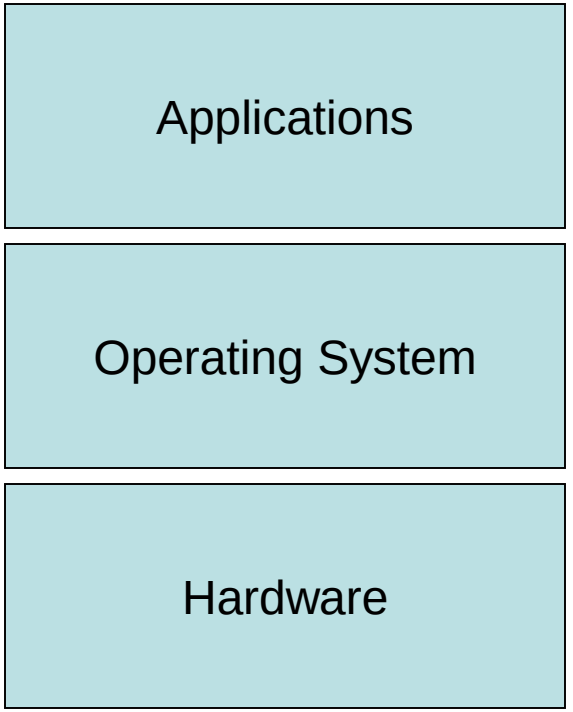

# Lecture 1
# Course Introduction and Organization

# CSCI 3753 Announcements

- Review of syllabus
- Introduction to Operating Systems
- Read chapters 1 and 2 in the textbook

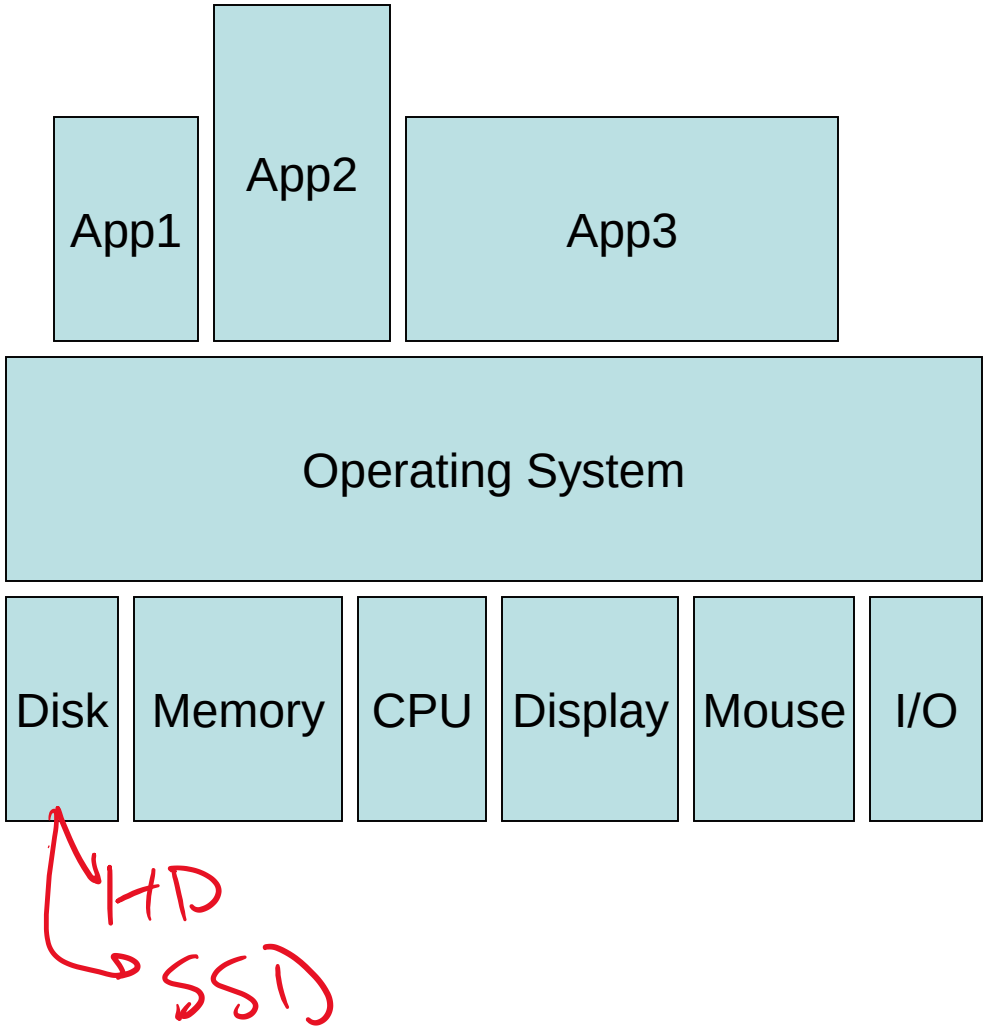
# What is an Operating System?

- Name some OSs: Windows, Linux, Mac OS X, Google Android, …
- What is common across these OSs?
- An operating system is a layer of software between applications and hardware



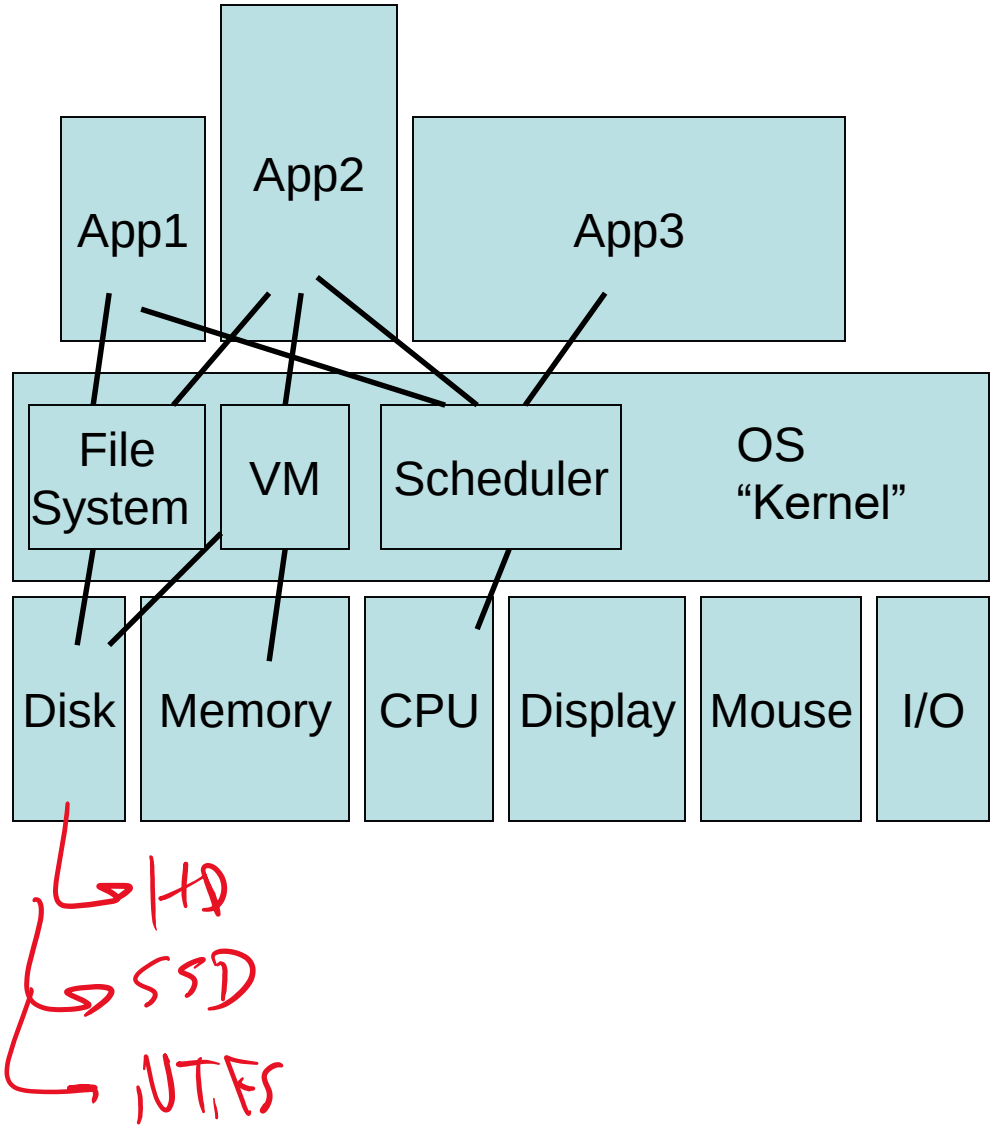
# What is an Operating System?

- An operating system is a layer of software between *many* applications and *diverse* hardware that
  - *Provides a hardware abstraction* so an application doesn't have to know details about the hardware.
    - otherwise an application saving a file to disk would have to know how the disk operates
  - Arbitrates access to resources among multiple applications:
    - *Sharing* of resources
    - *Isolation* protects app's from each other



# What is an Operating System?

- A PC operating system consists of multiple components
  - scheduler
  - virtual memory system
  - file system
  - device management
  - other...



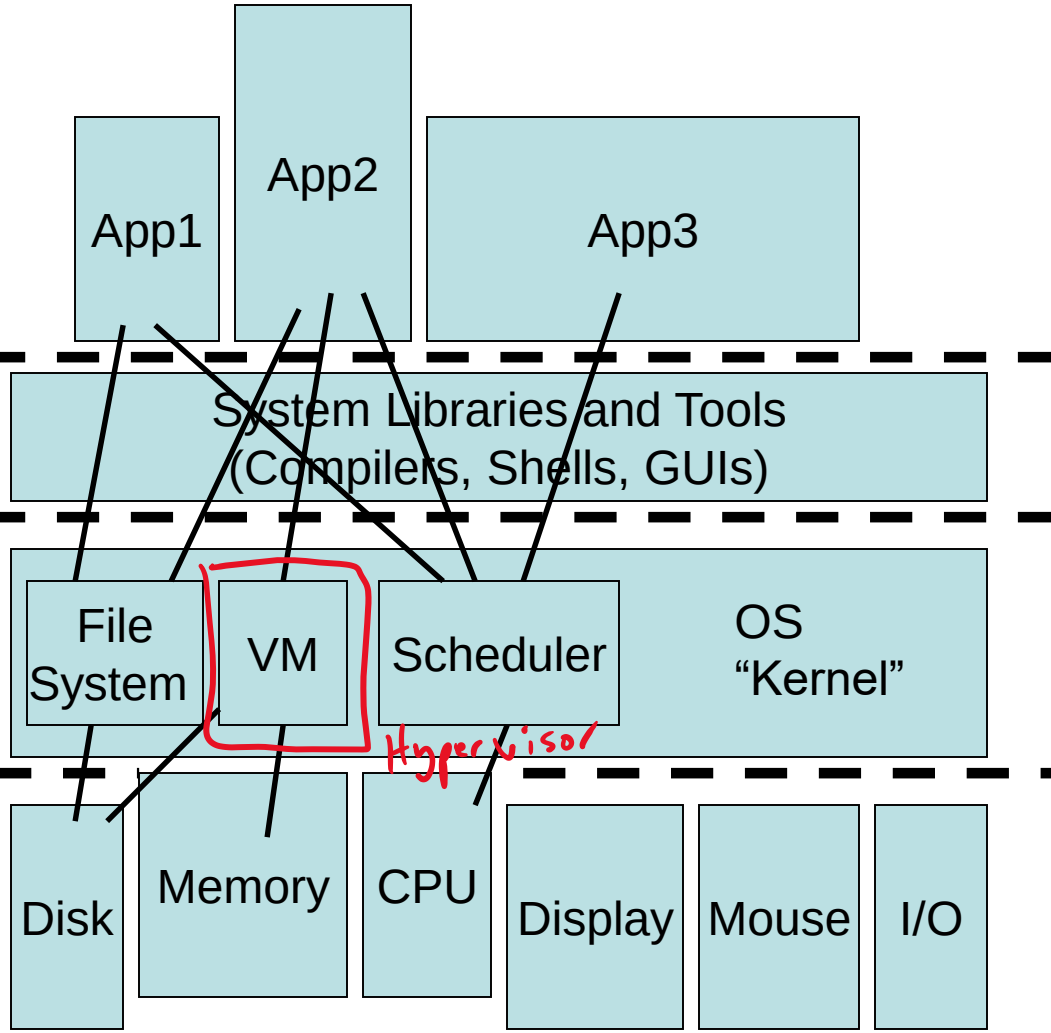
# What is an Operating System?

Posix, Win32,
Java, C library API

System call API

- 160 in Linux

Device driver "API"

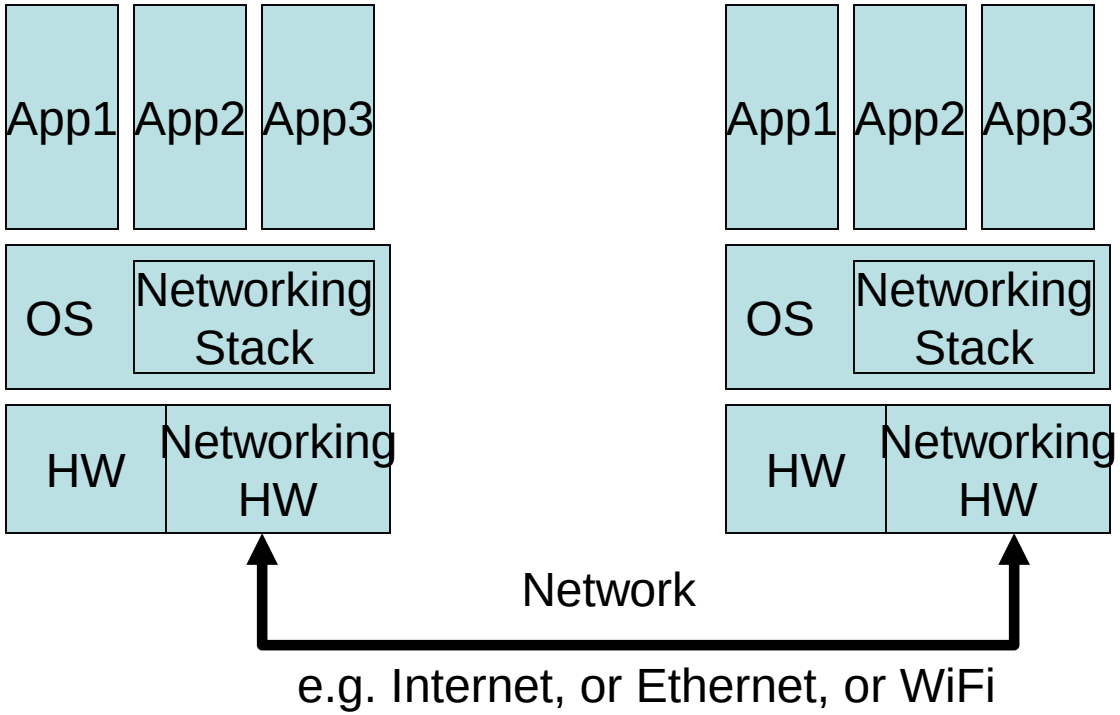



Note: different OS kernels can support the same system call API

# Outline of the OS course

1. Hardware supporting design, user/supervisor mode, system calls, trap table, device I/O, interrupts, DMA, mem-mapped I/O
2. Processes, threads, scheduling, synchronization, deadlock
3. Memory management, paging, virtual memory
4. File system design, allocation, networked file systems
5. Security: authorization, access control

# What is an Operating System?




- Examples:
  - App1 is a distributed client server app, e.g. App1 on left is Web browser, App1 on right is Web server

- Distributed Operating Systems
  - Networked File System
  - OS adds TCP/IP Network Stack
  - Device driver support for Networking cards

# Outline of the OS course

1. Hardware support, virtual machines, user/supervisor mode, system calls, trap table, device I/O, interrupts, DMA, memory-mapped I/O
2. Processes, threads, scheduling, synchronization, deadlock
3. Memory management, paging, virtual memory
4. File system design, allocation
5. Advanced Topics:

   Networked file systems

   Security: authorization, access control

# Summary…

- An OS is a software layer that sits between applications and I/O devices
  - Main Goals: Abstraction, Arbitration, & Protection
- An OS consists of many components
  - Memory manager, Scheduler, File System, Device Management, Network Stack, etc.
- Different OS Flavors have different design goals
  - Linux is a *monolithic* kernel:
    - complex, contains many components
  - Mach OS is a *microkernel*
    - kernel only contains scheduler, memory manager, and inter-processes communication (messaging)

# Questions?