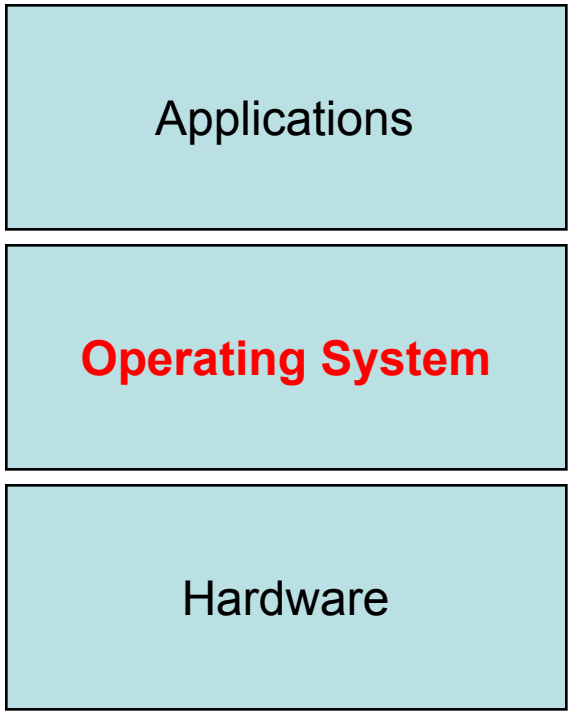
CSCI 3753 Operating Systems
Spring 2016

William Mortl, MS & Rick Han, PhD
Department of Computer Science
University of Colorado at Boulder

What is an Operating System?

- Name some OSs:
  - Windows, Linux, Mac OS X, Google Android, …
  - >500 at http://www.operating-system.org/betriebssystem/_english/os-liste.htm
  - >600 versions of Linux! i.e. over 600 distributions of Linux.
- What is common across these OSs?

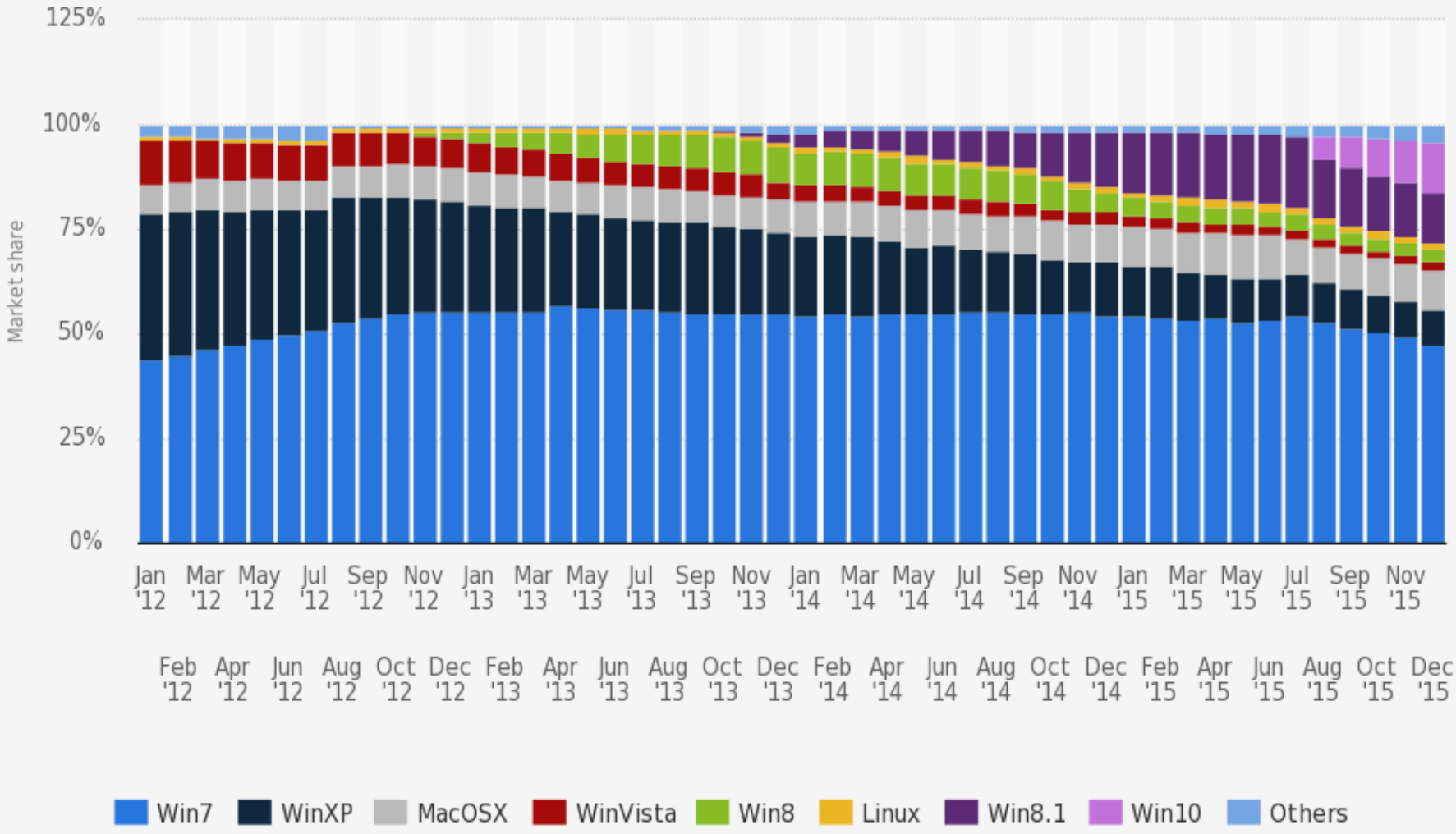


- An operating system is a layer of software between applications and hardware that provides useful services to applications

OS Historical Timeline

- 1930s/40s – electronic digital computers arise (ENIAC 1946 is 1st general-purpose programmable digital computer)
- 1950s – 1st OSs begin to emerge
- 1961 – MIT's CTSS is the first time-sharing system
- 1966 – IBM System/360 mainframe OS
- 1969 - UNIX for mainframes and minicomputers
- 1981 – MS-DOS OS for personal computers
- 1982 – 4.2 BSD Unix with TCP/IP networking
- 1984 – Mac OS with windowing GUI
- 1991 – Linux open source OS for PC
- Late 1990s/early 2000s – virtual machines arise like VMWare, Xen, …
- 2007 – iOS for iPhone

Global market share held by operating systems Desktop PCs from January 2012 to December 2015

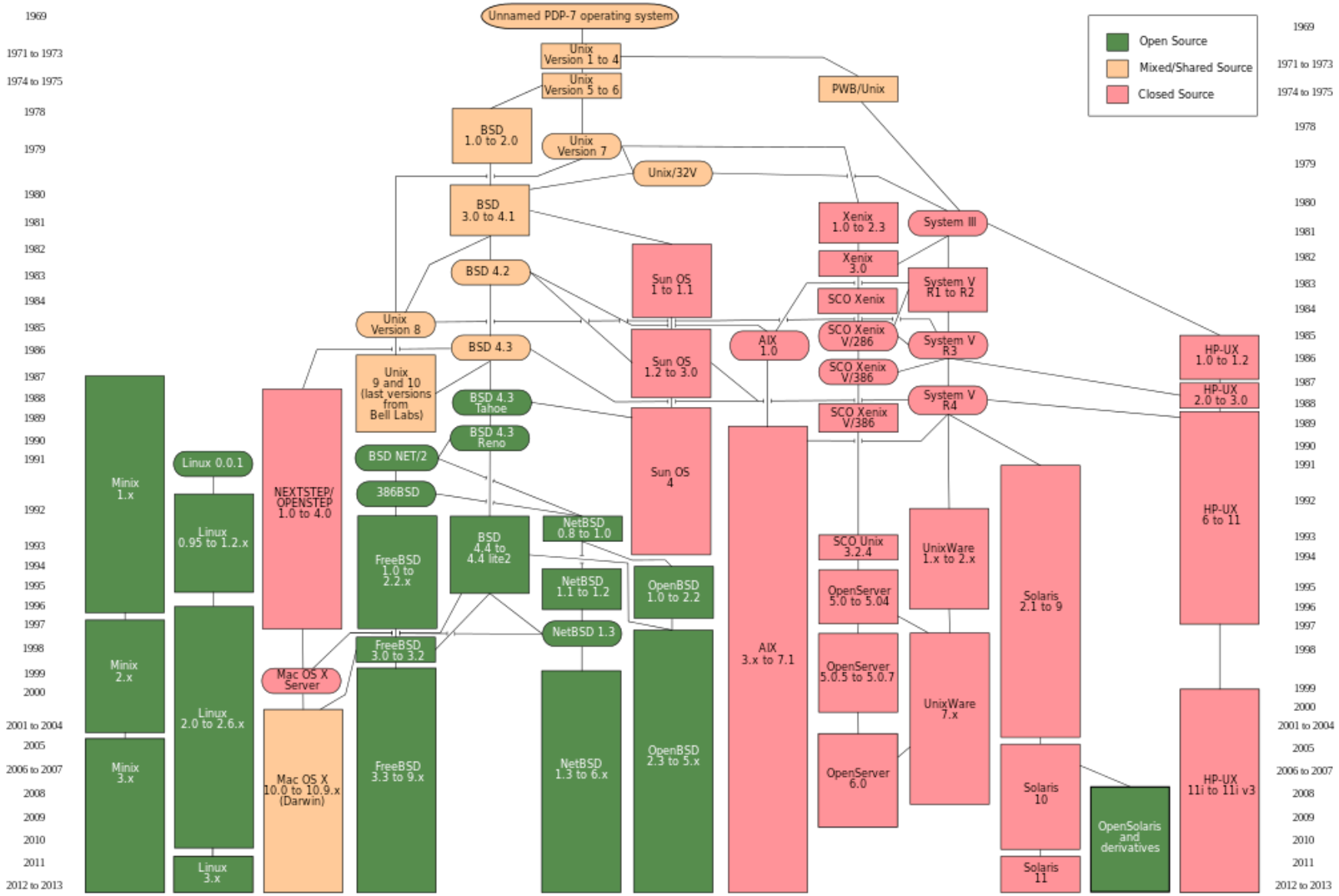


Source:
© Statista 2016

Additional Information:
Worldwide

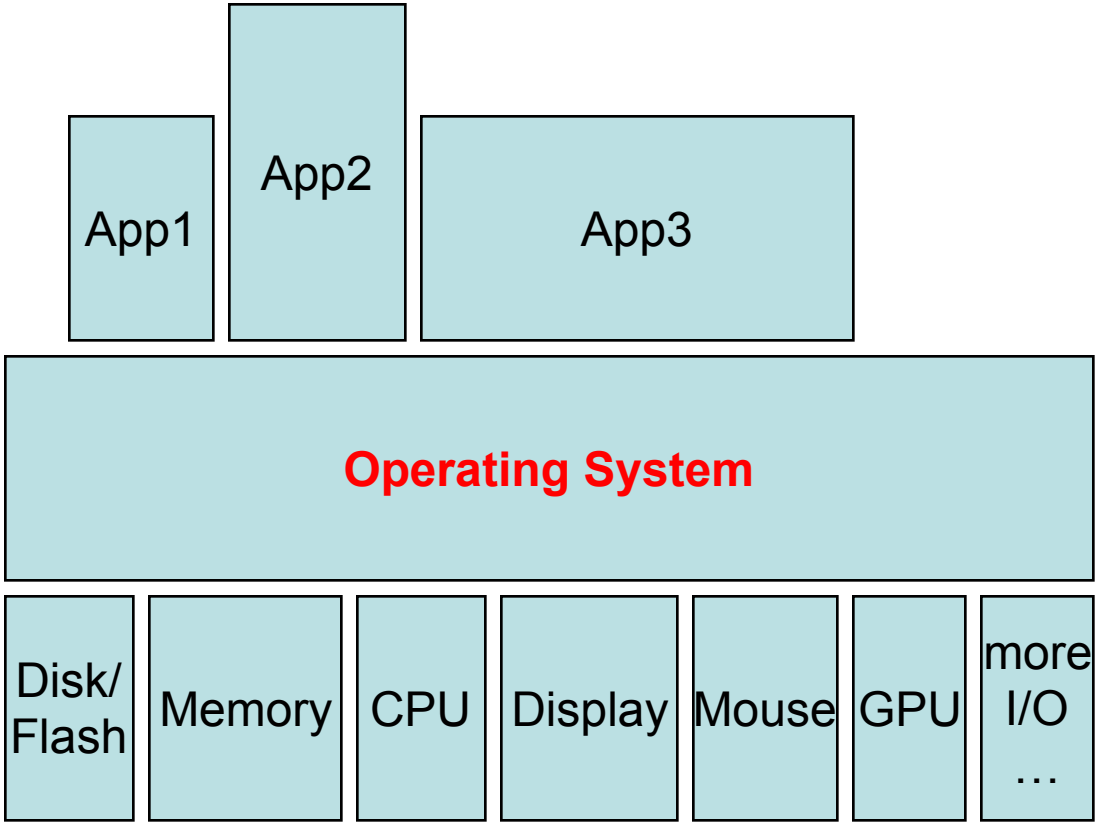
Windows approximately 86%

UNIX OS Family Tree



What is an Operating System? (2)

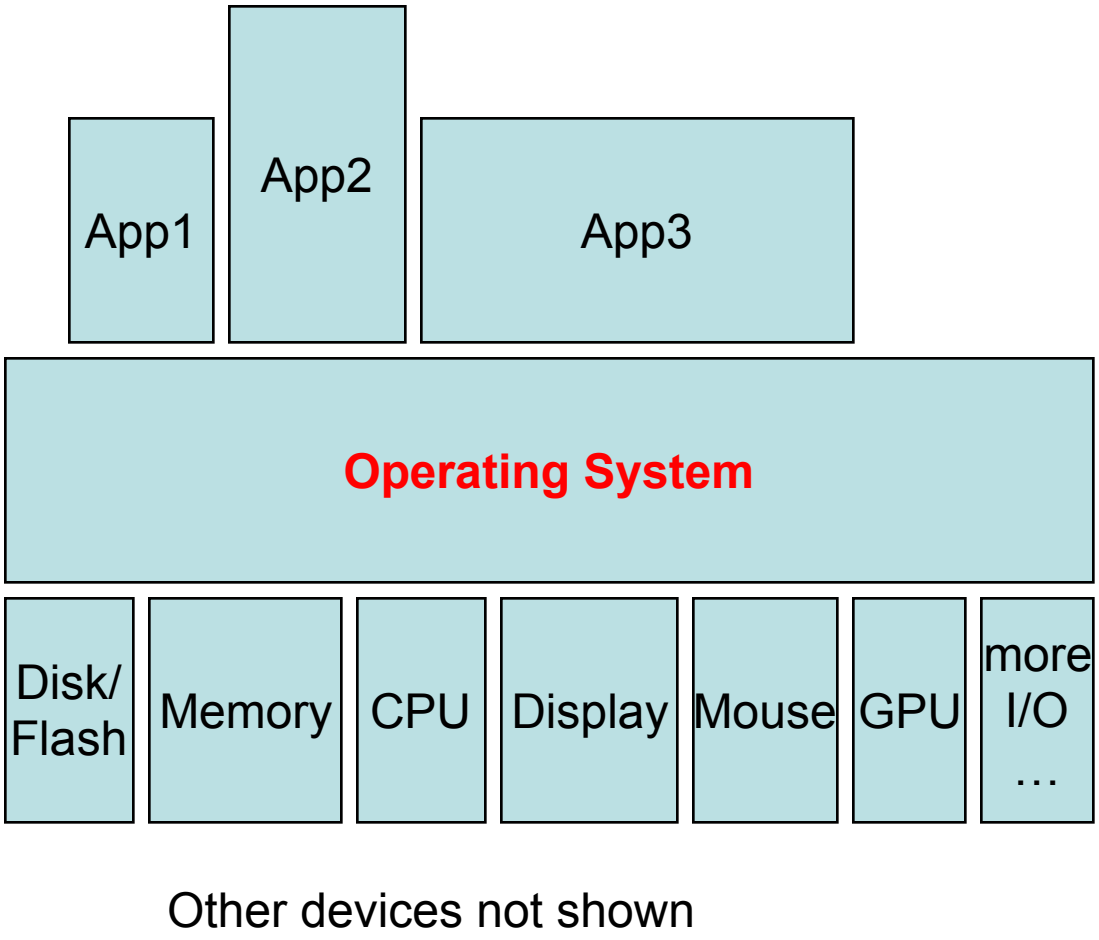
- Name some types of applications
  - Web browers, video players, games, Office apps, …
- Name different kinds of Input/Output (I/O) devices
  - Mouse, keyboard, display, disk, …



Other devices include: wired network card, WiFi, camera, microphone, audio output, keyboard, DVD/CD, USB, etc.

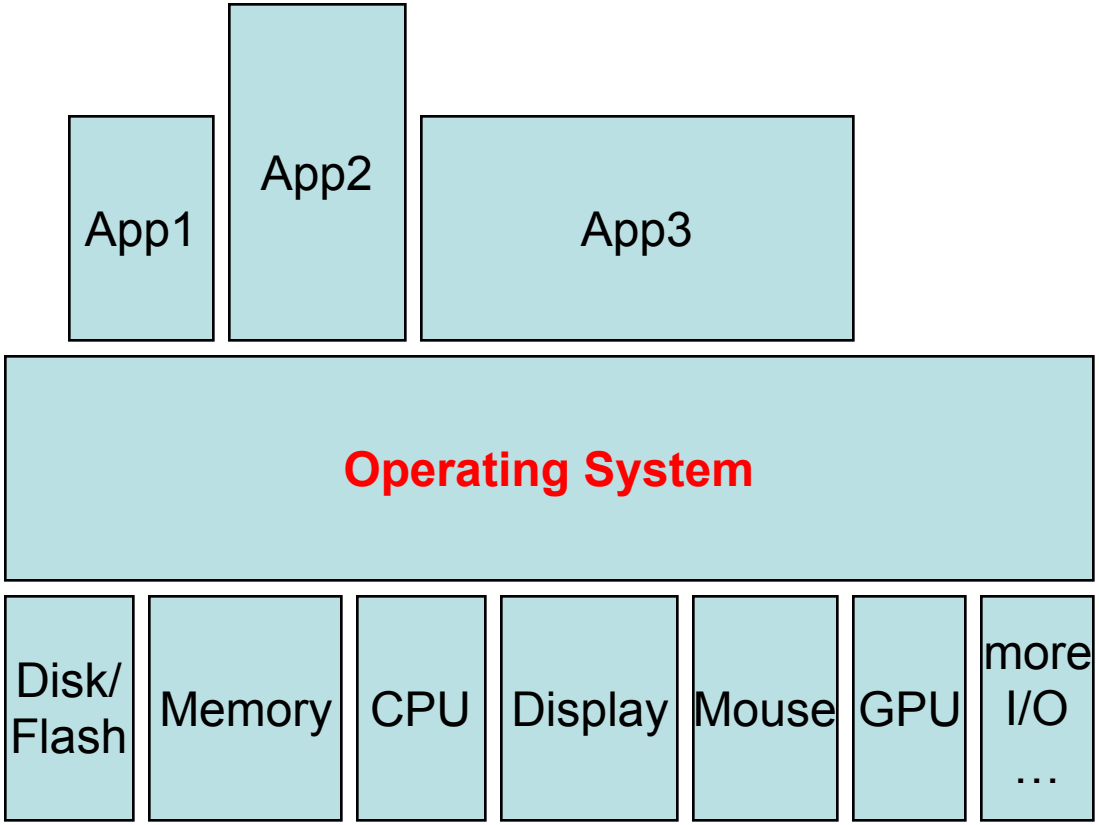
What is an Operating System? (3)

- An operating system is a layer of software between *many* applications and *diverse* hardware that
  1. *Provides a hardware abstraction* so an application doesn't have to know the details about the hardware.
     - So an application saving a file to disk doesn't have to know how the disk operates



What is an Operating System? (4)

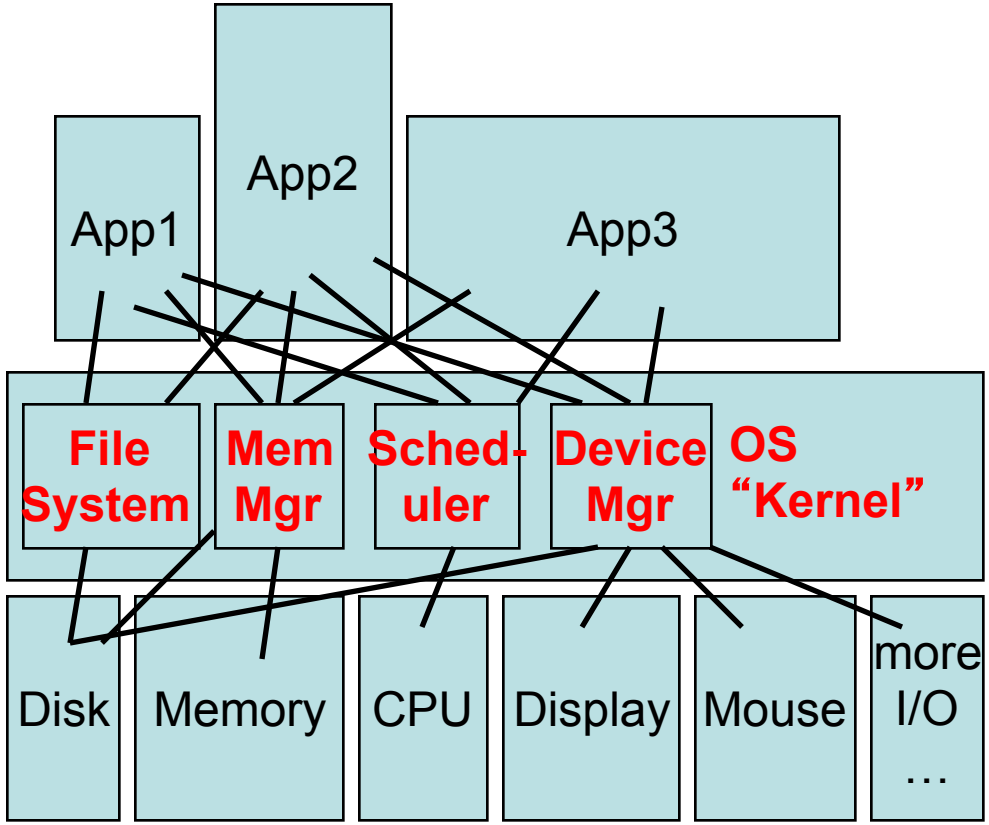
2. Arbitrates access to resources among multiple applications:
   - *Sharing* of resources
3. Provides Protection:
   - *Isolation* protects app's from each other
   - Isolation also to protect the OS from applications
   - Isolation to limit resource consumption by any one app



Other devices not shown

Operating System Components

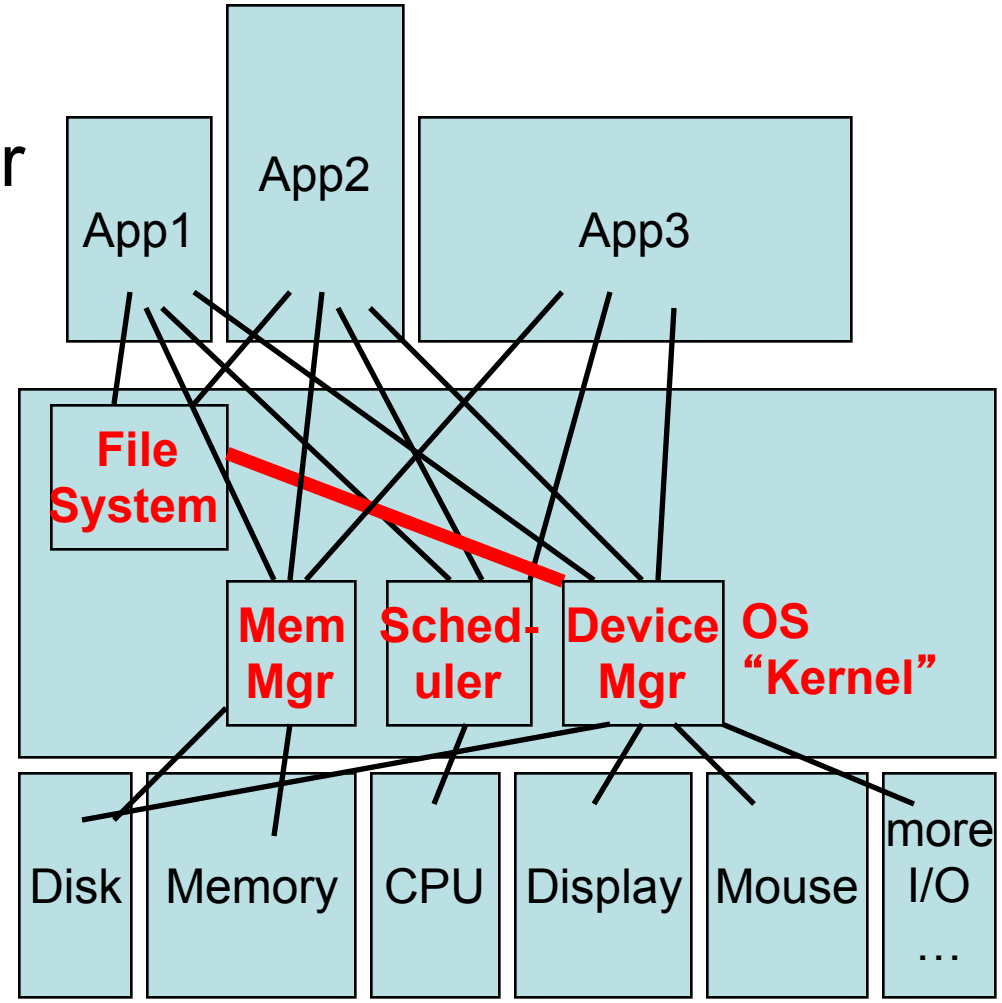
- A typical operating system consists of multiple components
  - A process manager with a scheduler, thread management, and atomic synchronization
  - Memory management (virtual memory) system
  - file system
  - device management
  - other...



Not pictured above in the OS: the network stack manager, etc.

Operating System Hierarchy

- OS components may be built on top of other OS components
  - The file system is usually built on top of the device manager
    - File system supports high level abstraction/concept of files
    - Device manager handles low level interaction with devices



Not pictured above in the OS: the network stack manager, etc.

ABI

- ABI – Application Binary Interface
  - Interface between a compiled executable and the operating system which loads it
  - Size, layout, and bit alignment of data types
  - Calling convention (how values are passed and returned, what register)
  - How an application makes system calls to the operating system
  - This is why applications compiled for Mac OSX will not run on Linux

API

- API – Application Programming Interface
  - Libraries, protocols, tools, functions, and objects for building programs and applications
  - Operating systems export an API, as do various add on libraries and toolkits (databases, hardware devices, etc.)
  - An API can be web based via web service calls, track a package on UPS:

https://www.ups.com/content/us/en/tracking/tools/index.html?WT.svl

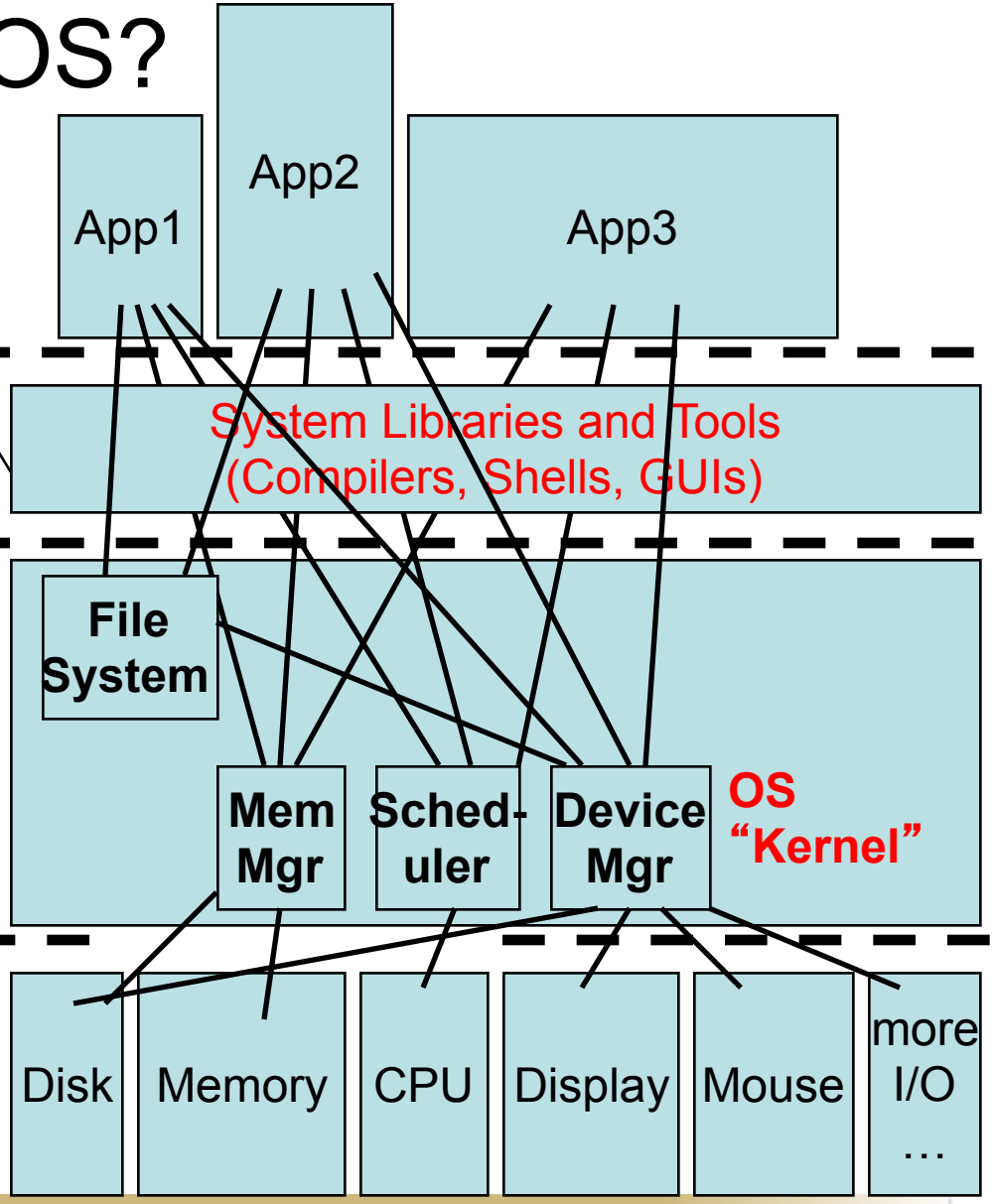
Are System Libraries/Tools part of an OS?

Posix, Win32,
Java, C library API

System call API

– >300 in Linux 3.1

Device driver "API"

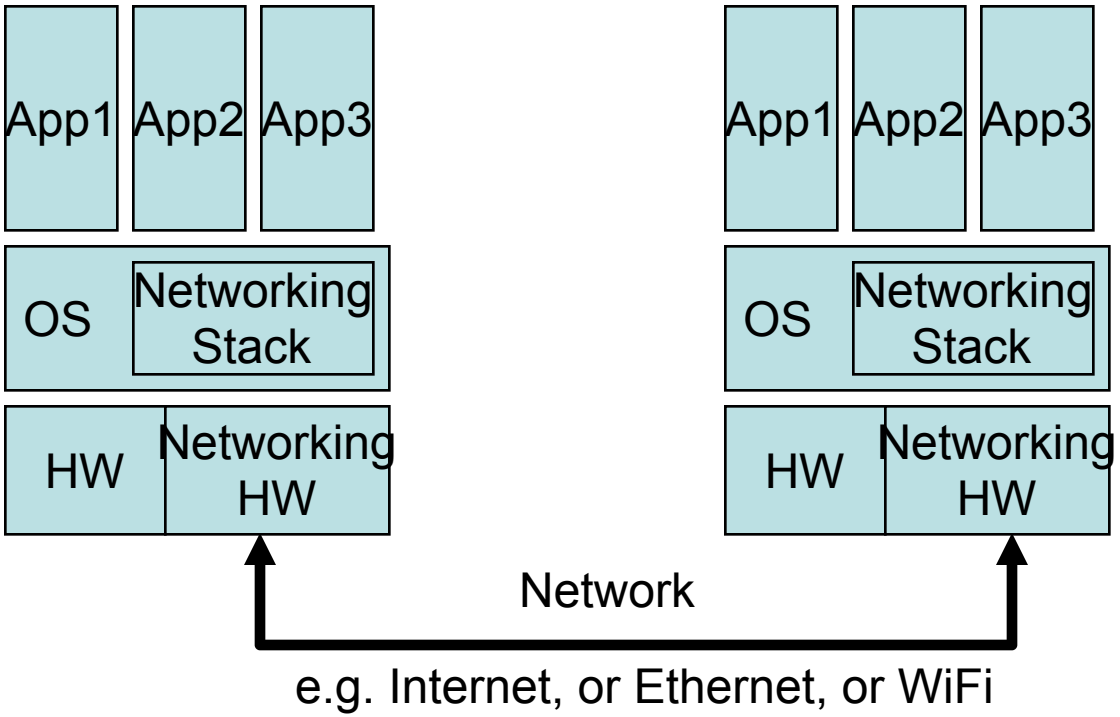


Note: different OS kernels can support the same system call API

What's in an OS? Monolithic vs Microkernel OS Architectures

- Linux has a monolithic kernel
  - The kernel is highly complex and contains many components
- Mach OS has a *microkernel*
  - The kernel has minimal functionality, perhaps only virtual memory, scheduler, and IPC message passing
    - All other components are viewed as outside the OS, and communicate via message passing.
  - Advantage: bug in a component doesn't crash whole kernel, arguably easier to manage
  - Disadvantage: message passing was slow
- Mac OSX is a hybrid of Mach and BSD Unix

Distributed Operating Systems



- Example 1: Distributed File System
  - OS adds TCP/IP Network Stack
  - Device driver support for networking cards
  - Files can now be written/read remotely

- Example 2:
  - App1 is a distributed client server app, e.g. App1 on left is Web browser, App1 on right is Web server

Outline of the OS course

1. Hardware support, 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
6. Networked OSs: distributed file systems
7. Virtual machines
8. Network programming APIs
9. Windows / Mac OSX / iOS / Android Architecture (time permitting)

Operating System Trends

- Hardware has evolved quickly - OS must adjust
  - Moore's Law roughly applies to CPU speed and/or memory size: doubles every 18 months => exponential!
    - Enables complex modern operating systems like Linux, Windows, UNIX, OS X

Storage Evolution:

Punch card stores code 1950s-80s

Tape Drive 1950s-80s

Disk Drive 1960s-2000s

??

Flash Drive already at 32 GB @ $16=> OS disk scheduling obsolete??

But Moore's Law doesn't apply to disk access speed or to battery life

Operating System Trends

- Diversification of OS's to many different target environments
  - Energy-efficient cell phone OSs - scaling down
    - iPhone's iOS, Google's Android, …
  - Multi-processor OSs - scaling up
    - Adapting Linux and Windows to multiple cores. Massively parallel supercomputers.
  - Real-Time OS for Embedded and Multimedia Systems
    - VXWorks, robotic OSs, …

Operating System Trends

- Virtualization – Virtual Machines (VMs)
  - Running a Windows VM inside a Linux OS, and vice versa.
  - More layers of abstraction
- Cloud computing rents VMs on racks of PCs at a massive scale

Google Data Center in The Dalles, Oregon

Cloud Computing