Operating Systems: Principles and Practice

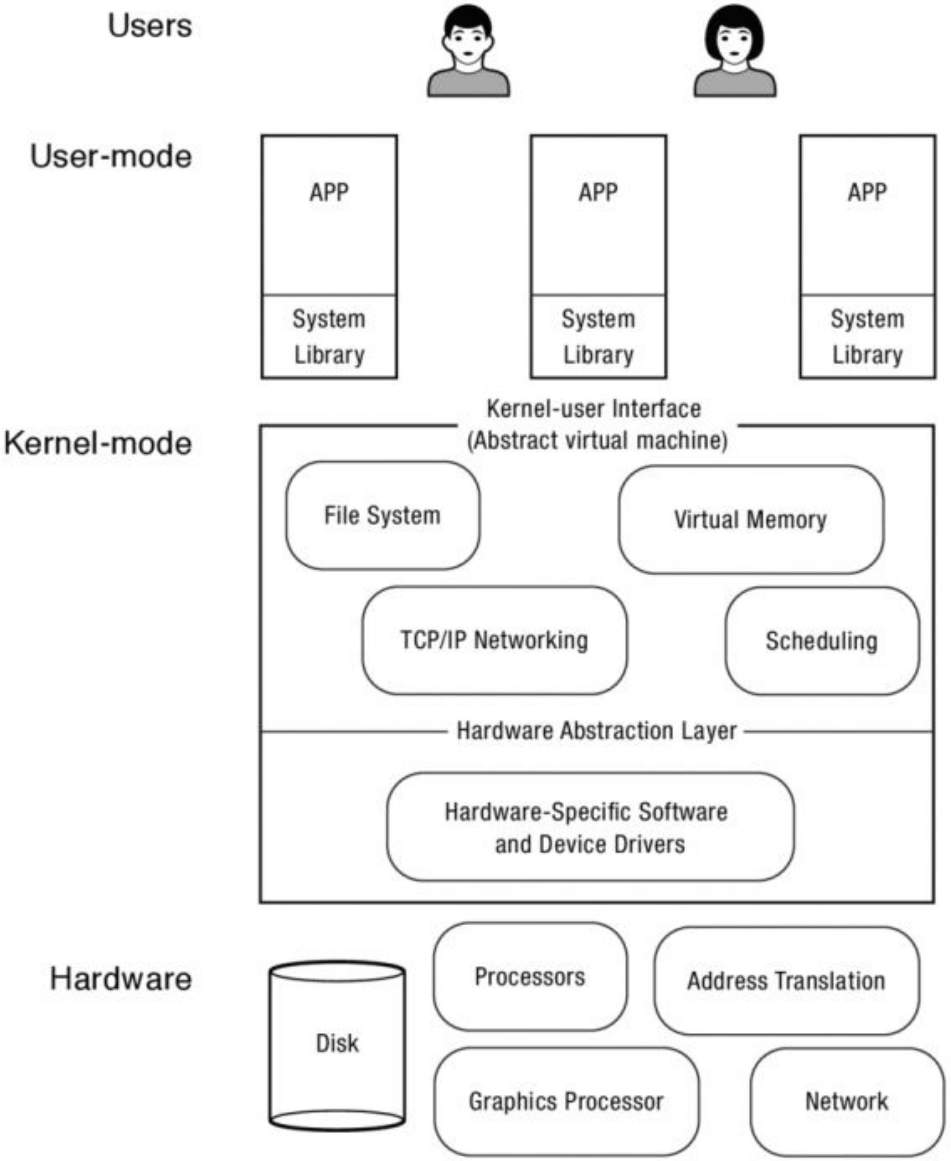
Rodrigo Gonzalez

Main Points (for today)

- Operating system definition
  - Software to manage a computer’s resources for its users and applications
- OS challenges
  - Reliability, security, responsiveness, portability, …
- OS history
  - How are OS X, Windows 8, and Linux related?

What is an operating system?

- Software to manage a computer's resources for its users and applications



Operating System Roles

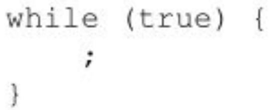
- Referee:
  - Resource allocation among users, applications
  - Isolation of different users, applications from each other
  - Communication between users, applications
- Illusionist
  - Each application appears to have the entire machine to itself
  - Infinite number of processors, (near) infinite amount of memory, reliable storage, reliable network transport
- Glue
  - A set of common services that facilitate sharing among applications
  - Libraries, user interface widgets, …

Example: File Systems

- Referee
  - Prevent users from accessing each other's files without permission
  - Even after a file is deleting and its space re-used
- Illusionist
  - Files can grow (nearly) arbitrarily large
  - Files persist even when the machine crashes in the middle of a save
- Glue
  - Named directories, printf, …

Operating System as Referee

- **Resource allocation**. The operating system must keep all simultaneous activities separate, allocating resources to each as appropriate



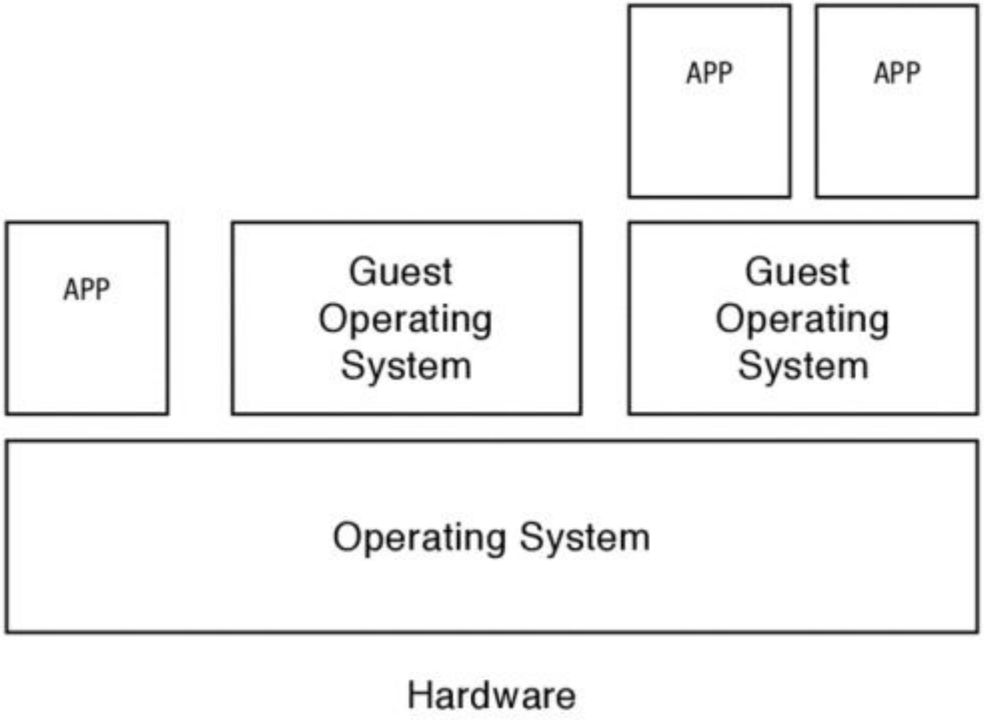
```
while (true) {
        ;
}
```

- **Isolation.** An error in one application should not disrupt other applications, or even the operating system itself.
- **Communication.** The flip side of isolation is the need for communication between different applications and different users

Operating System as Illusionist

- A second important role of an operating system is to mask the restrictions inherent in computer hardware
- Virtualization provides an application with the illusion of resources that are not physically present
  - Illusion of a nearly inf
  - OS masks network fai
    reliable service.
  -

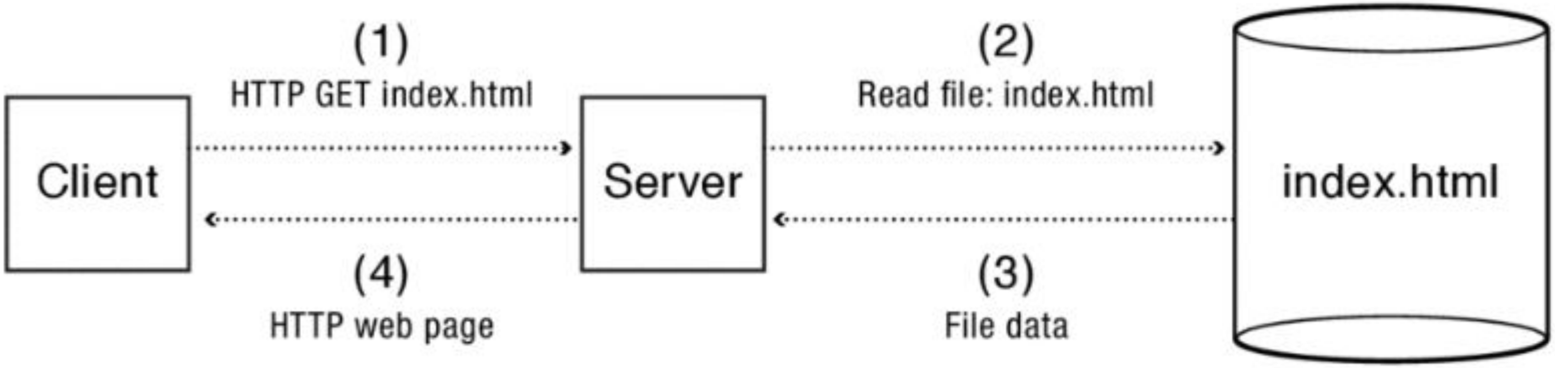
○



Operating System as Glue

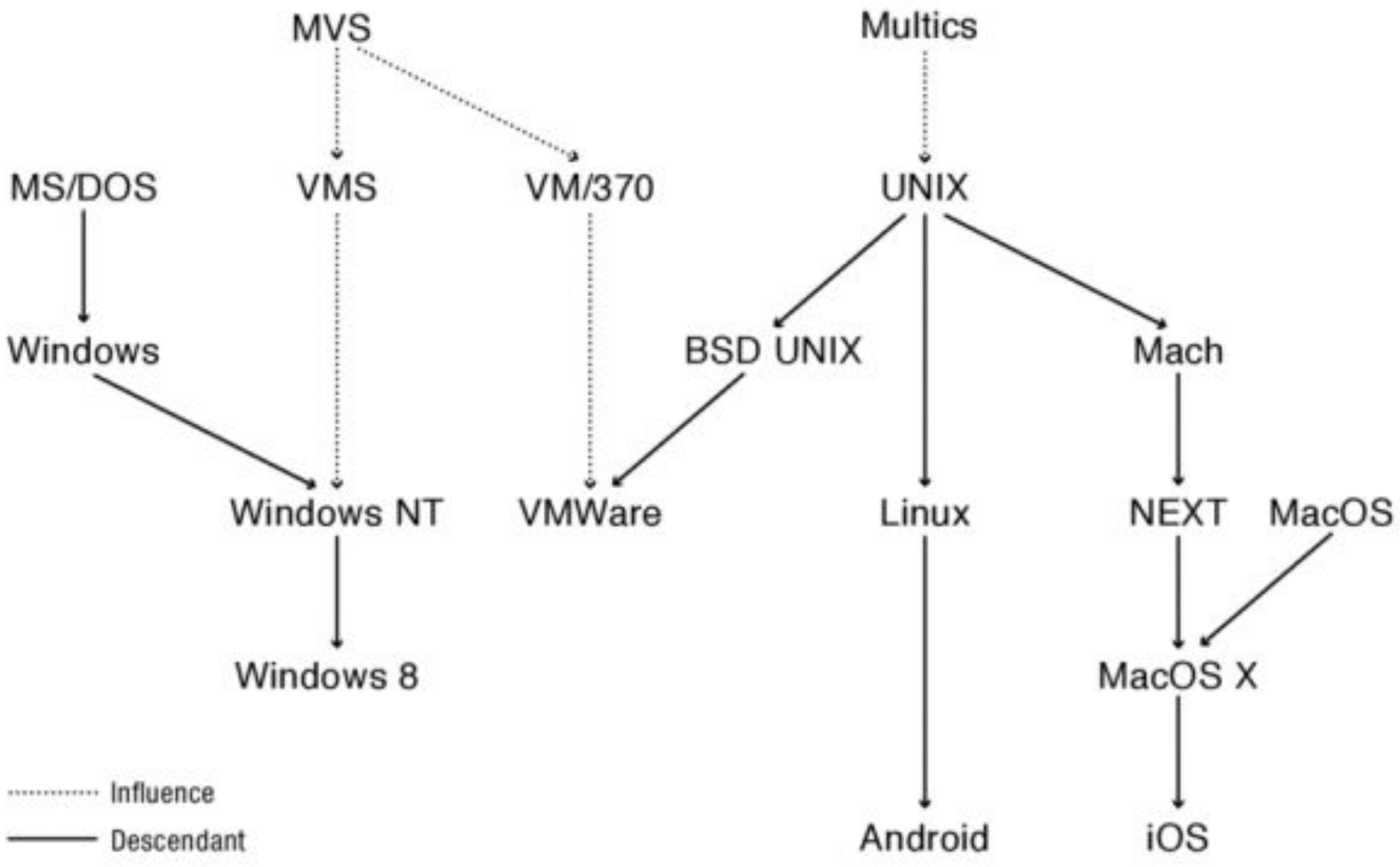
- OS provides a set of common, standard services to applications to simplify and standardize their design.
- The operating system hides the specifics of how the network and disk devices work, providing a simpler abstraction based on receiving/sending reliable streams of bytes and reading/writing named files
- For applications to share files, they must be stored in a standard format
- It provide a standard way for applications to pass messages and to share memory.

Example: web service

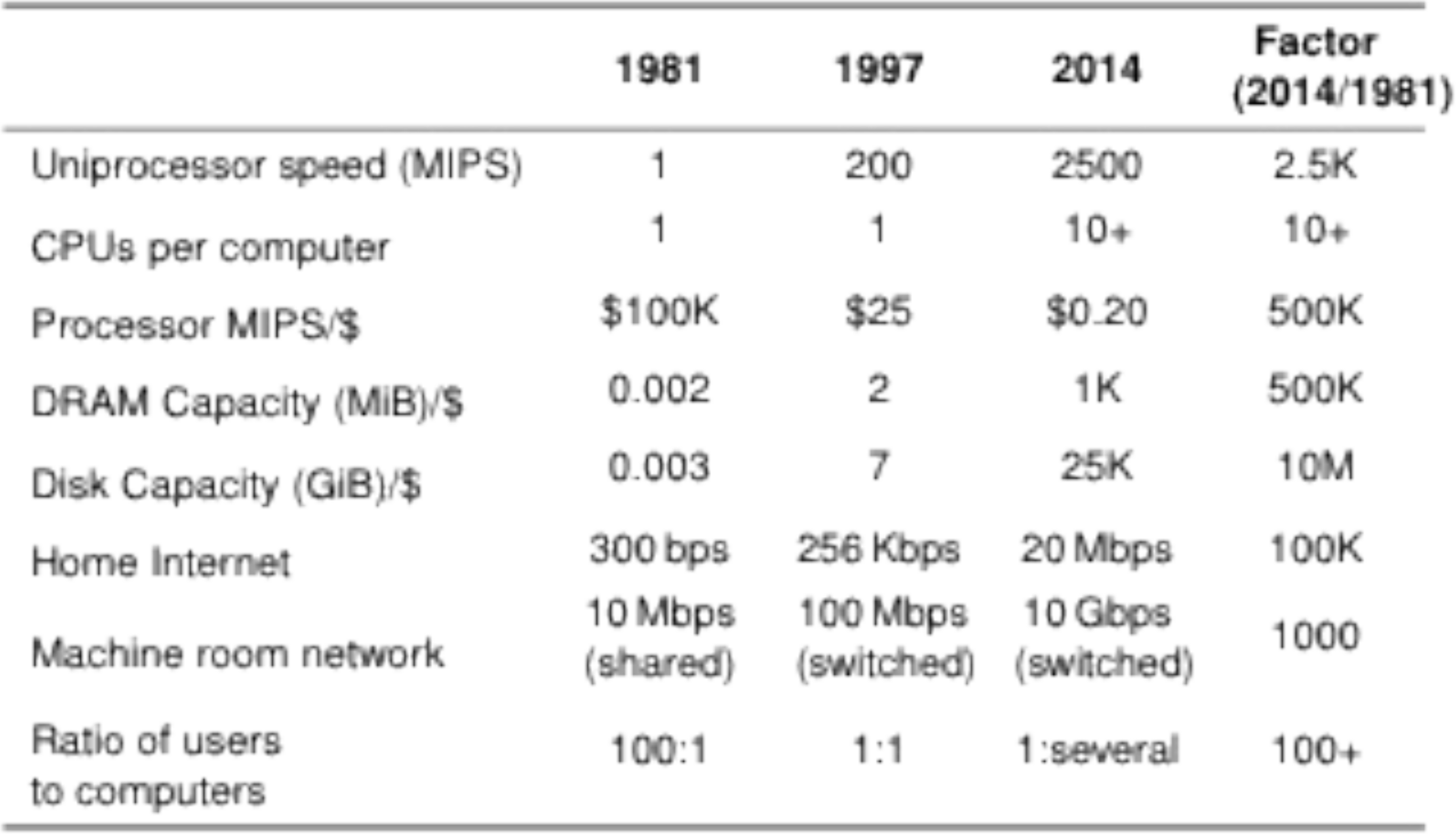


- How does the operating system enable multiple applications to communicate
- with each other?
- How does the operating system enable applications to do multiple things at once?
- How does the operating system synchronize application access to shared data?
- Does the operating system be re-written for every new piece of hardware?

OS History


MVS
Multics
MS/DOS
VMS
VM/370
UNIX
Windows
BSD UNIX
Mach
Windows NT
VMWare
Linux
NEXT
MacOS
Windows 8
MacOS X
Android
iOS
Influence
Descendant

Computer Performance Over Time



	1981	1997	2014	Factor (2014/1981)
Uniprocessor speed (MIPS)	1	200	2500	2.5K
CPUs per computer	1	1	10+	10+
Processor MIPS/$	$100K	$25	$0.20	500K
DRAM Capacity (MiB)/$	0.002	2	1K	500K
Disk Capacity (GiB)/$	0.003	7	25K	10M
Home Internet	300 bps	256 Kbps	20 Mbps	100K
Machine room network	10 Mbps (shared)	100 Mbps (switched)	10 Gbps (switched)	1000
Ratio of users to computers	100:1	1:1	1:several	100+

Early Operating Systems: Computers Very Expensive

- One application at a time
  - Had complete control of hardware
  - OS was runtime library
  - Users would stand in line to use the computer
- Batch systems
  - Keep CPU busy by having a queue of jobs
  - OS would load next job while current one runs
  - Users would submit jobs, and wait, and wait, and

Time-Sharing Operating Systems: Computers and People Expensive

- Multiple users on computer at same time
  - Multiprogramming: run multiple programs at same time
  - Interactive performance: try to complete everyone’s tasks quickly
  - As computers became cheaper, more important to optimize for user time, not computer time

Today’s Operating Systems: Computers Cheap

- Smartphones
- Embedded systems
- Laptops
- Tablets
- Virtual machines
- Data center servers

Tomorrow’s Operating Systems

- Giant-scale data centers
- Increasing numbers of processors per computer
- Increasing numbers of computers per user
- Very large scale storage

Textbook

Thomas Anderson and Mike Dahlin Operating Systems Principles & Practice Volume I: Kernels and Processes, Second Edition. Chapter 1.