CS 423
Operating System Design: Overview and Basic Concepts

Professor Adam Bates
Fall 2018

Goals for Today

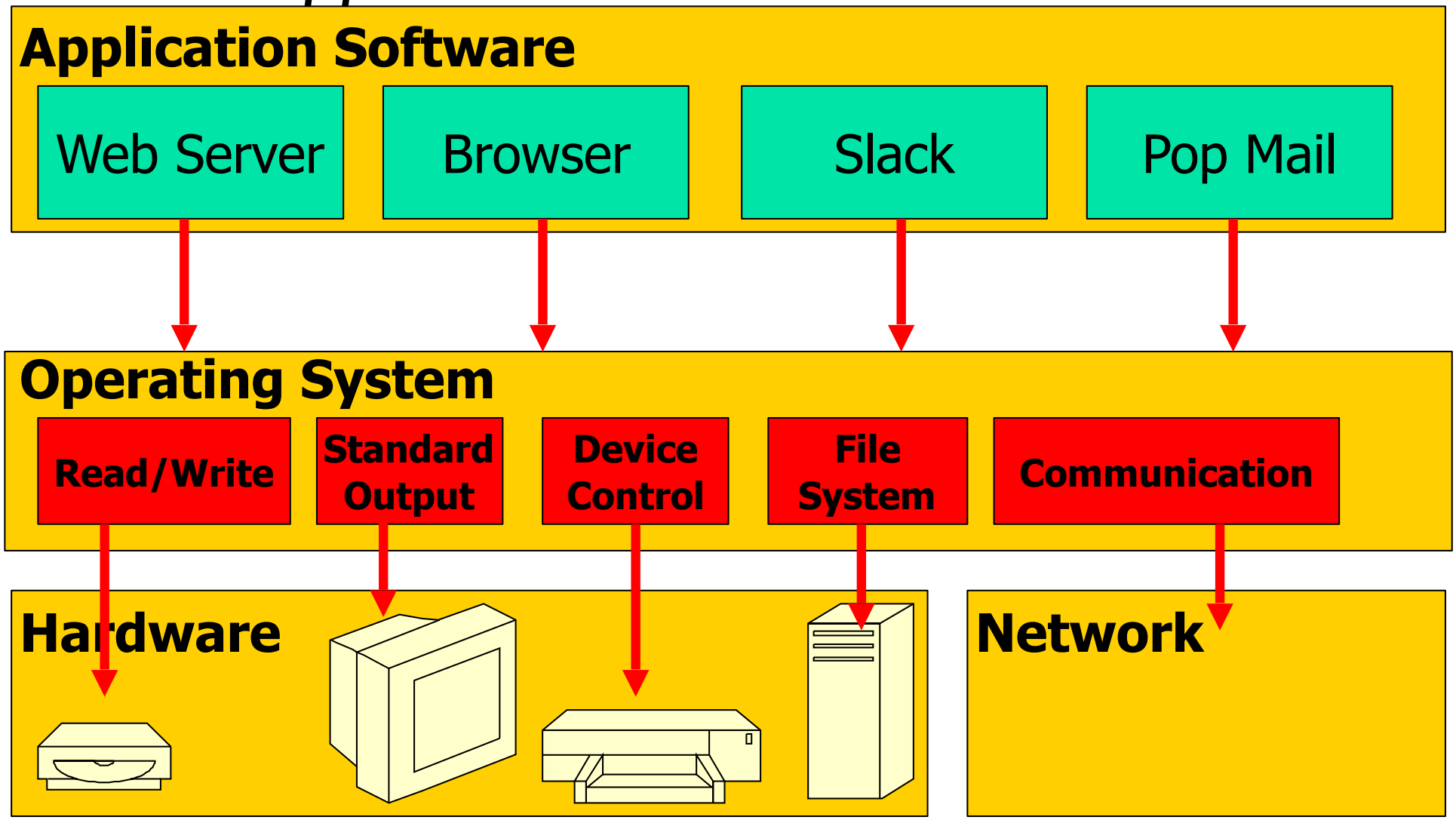
- Learning Objectives:
  - Introduce OS definition, challenges, and history
- Announcements:
  - C4 readings for Week 2 are out! **Due Jan 26**
  - MP0 is available on Compass! **Due Jan 29**
  - HW0 is available on Compass! **Due Jan 29**

Reminder: Please put away devices at the start of class

What is an operating system?

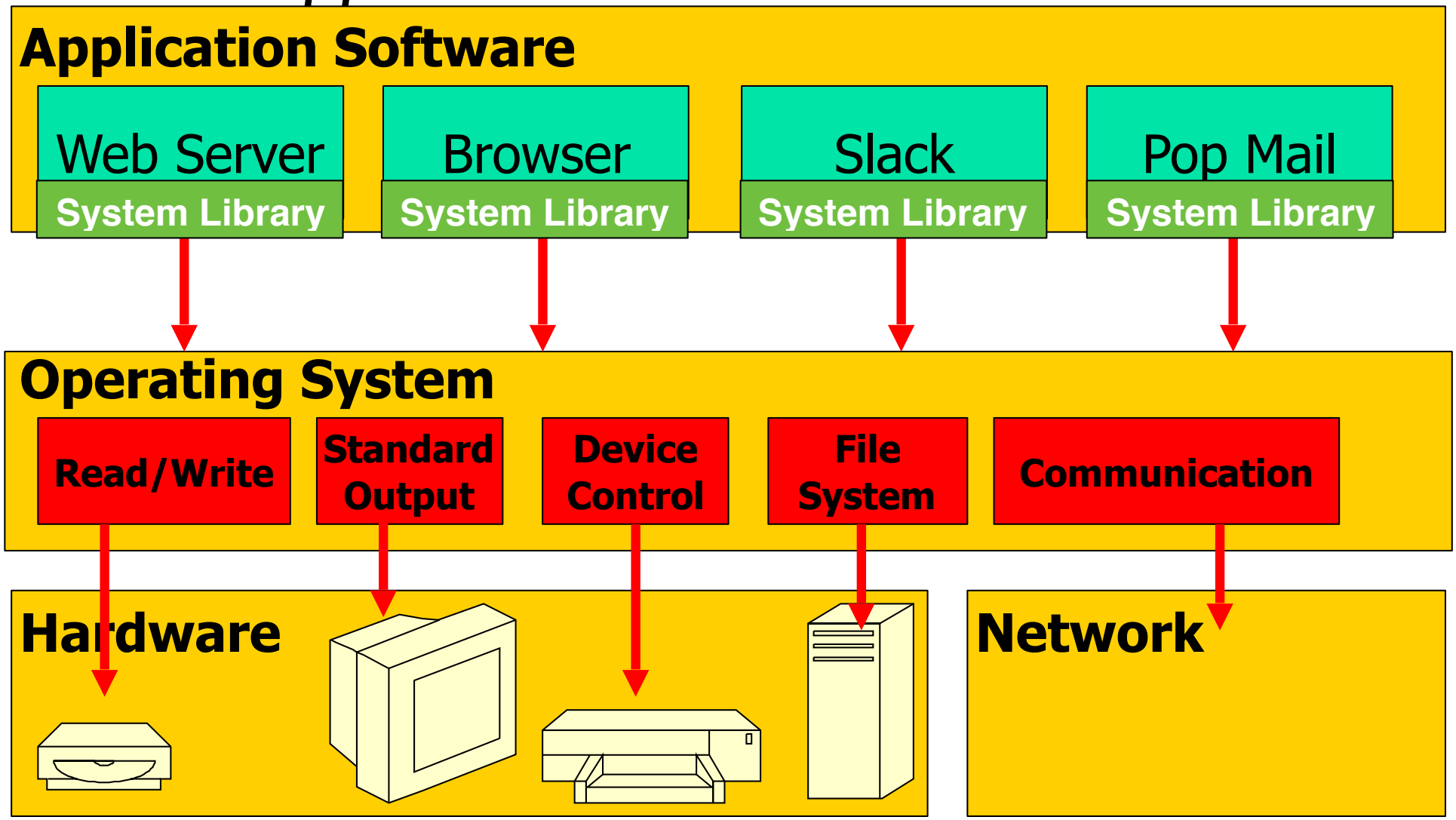
Why Operating Systems?

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



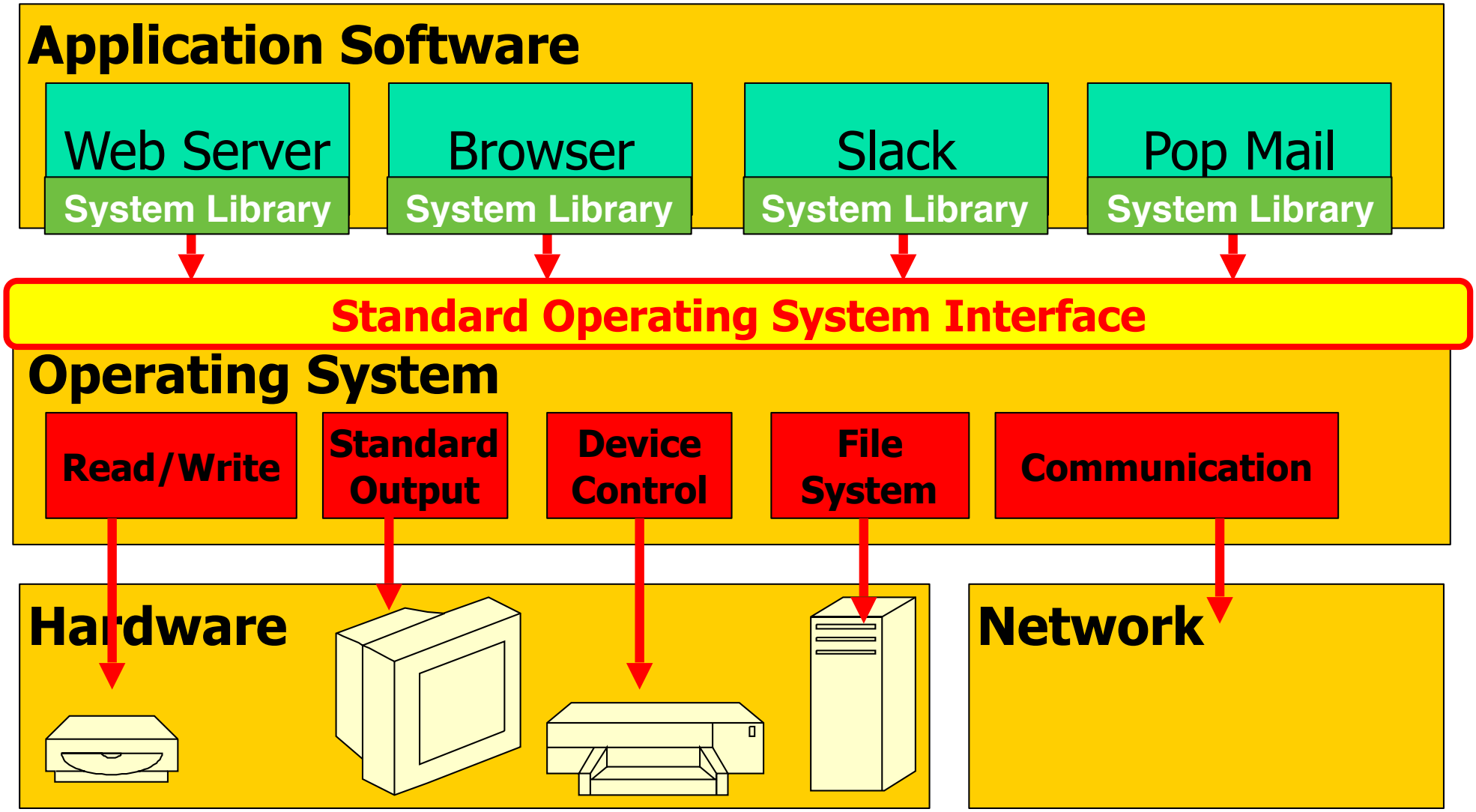
Why Operating Systems?

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



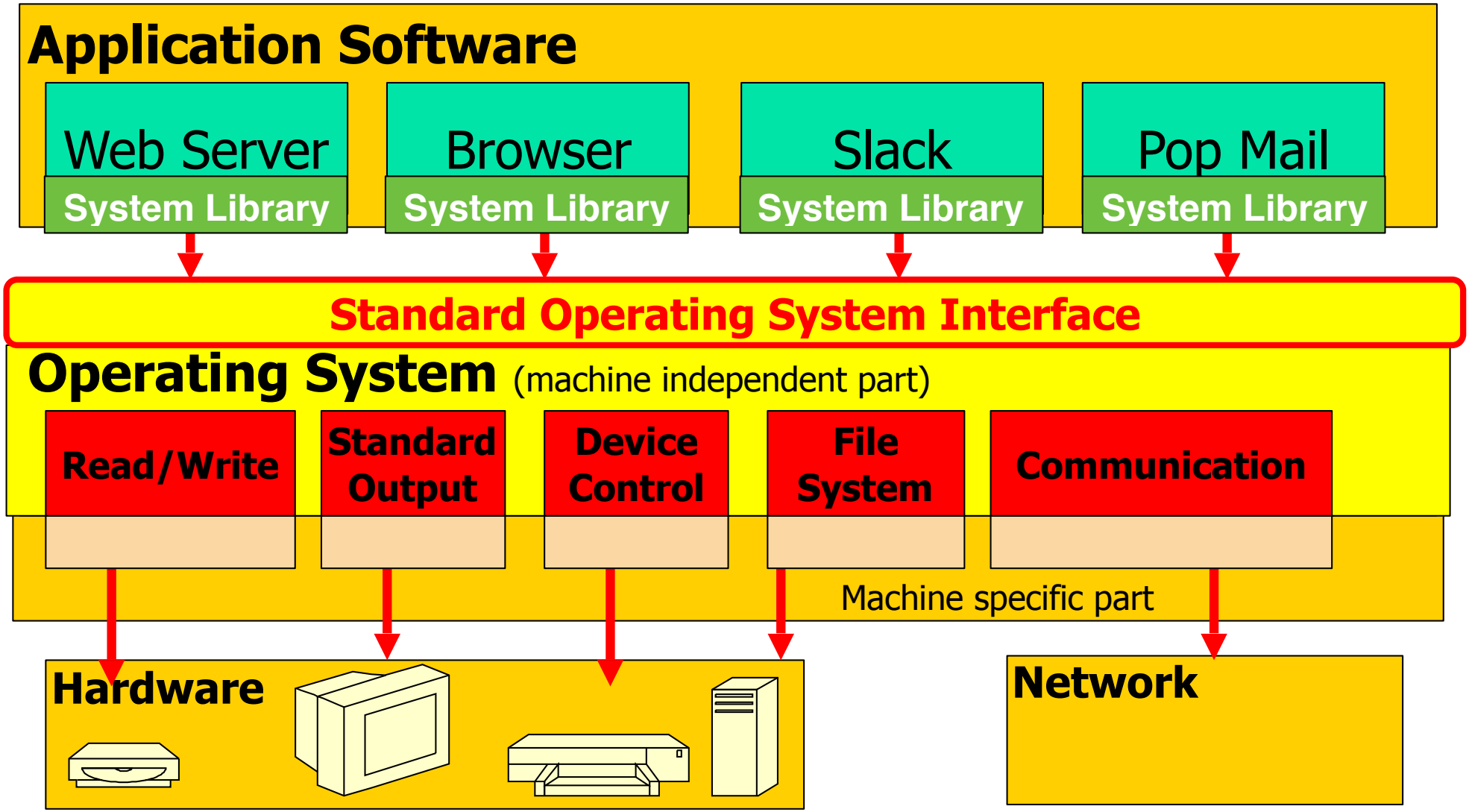
Why Operating Systems?

The OS exports a user interface. Why?



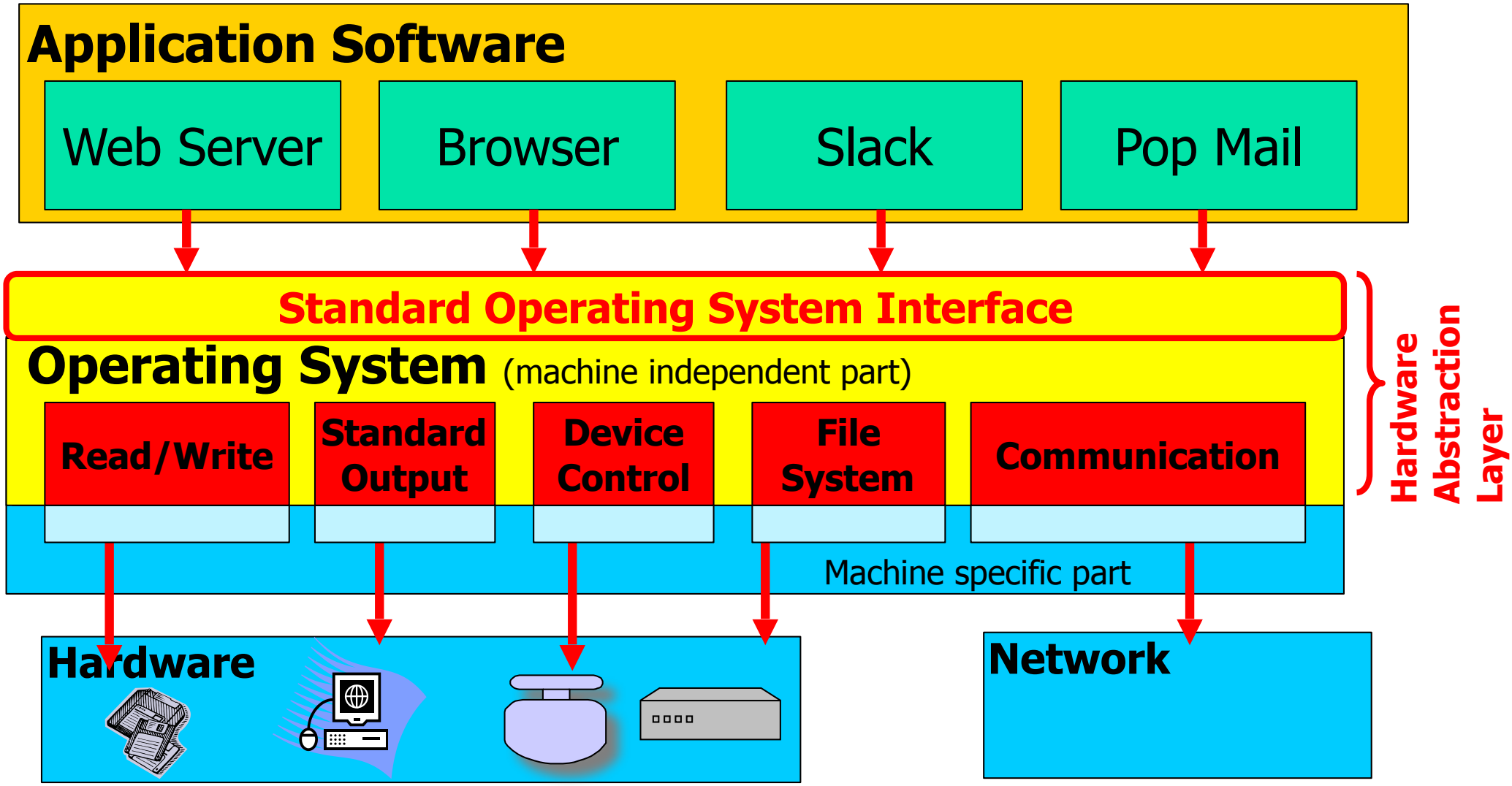
Why Operating Systems?

Standard interface increases portability and reduces the need for machine-specific code.



Why Operating Systems?

OS Runs on Multiple Platforms while presenting the same Interface:



What are the responsibilities of an operating system?

Operating System Roles

Role #1: Referee

- Manage resource allocation between users and applications
- Isolate different users and applications from one another
- Facilitate and mediate communication between different users and applications

Operating System Roles

Role #2: Illusionist

- Allow each application to believe it has the entire machine to itself
- Create the appearance of an Infinite number of processors, (near) infinite memory
- Abstract away complexity of reliability, storage, network communication…

Operating System Roles

Role #3: Glue

- Manage hardware so applications can be machine-agnostic
- Provide a set of common services that facilitate sharing among applications
- **Examples of "Glue" OS Services?**

Operating System Roles

Role #3: Glue

- Manage hardware so applications can be machine-agnostic
- Provide a set of common services that facilitate sharing among applications
- **Examples of "Glue" OS Services?**
  - Cut-and-paste, File I/O, User Interfaces...

Consider file systems and storage devices…

How is the OS a referee?
An illusionist?
Glue?

Ex: File System Support

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, other system calls for File I/O

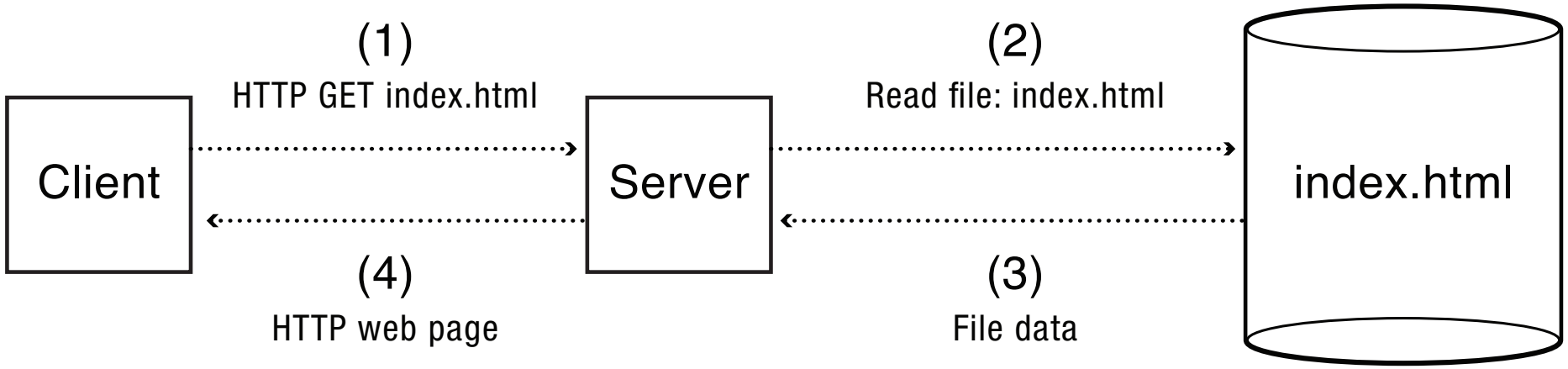
A Question

What does an OS need to do in order safely run an untrustworthy application?

Another Question

How should an operating system allocate processing time between competing uses?

Example: Web Service



- How does the server manage many simultaneous client requests?
- How do we keep the client safe from spyware embedded in scripts on a web site?
- How do handles updates to the web site such that clients always see a consistent view?

OS Challenges

Reliability

- Does the system do what it was designed to do?

Availability

- What portion of the time is the system working?
- Mean Time To Failure (MTTF), Mean Time to Repair

Security

- Can the system be compromised by an attacker?

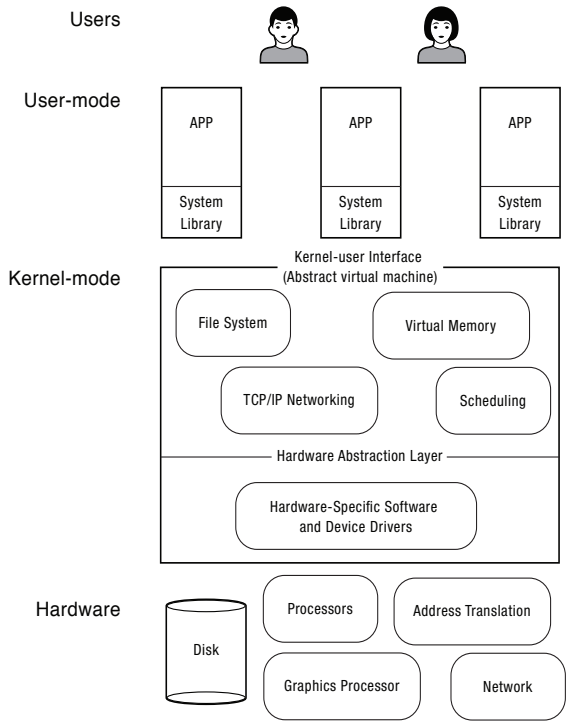
Privacy

- Data is accessible only to authorized users

OS Challenges

Portability

- For programs:
  Application programming interface (API)
  Abstract virtual machine (AVM)
- For hardware
  Hardware abstraction layer

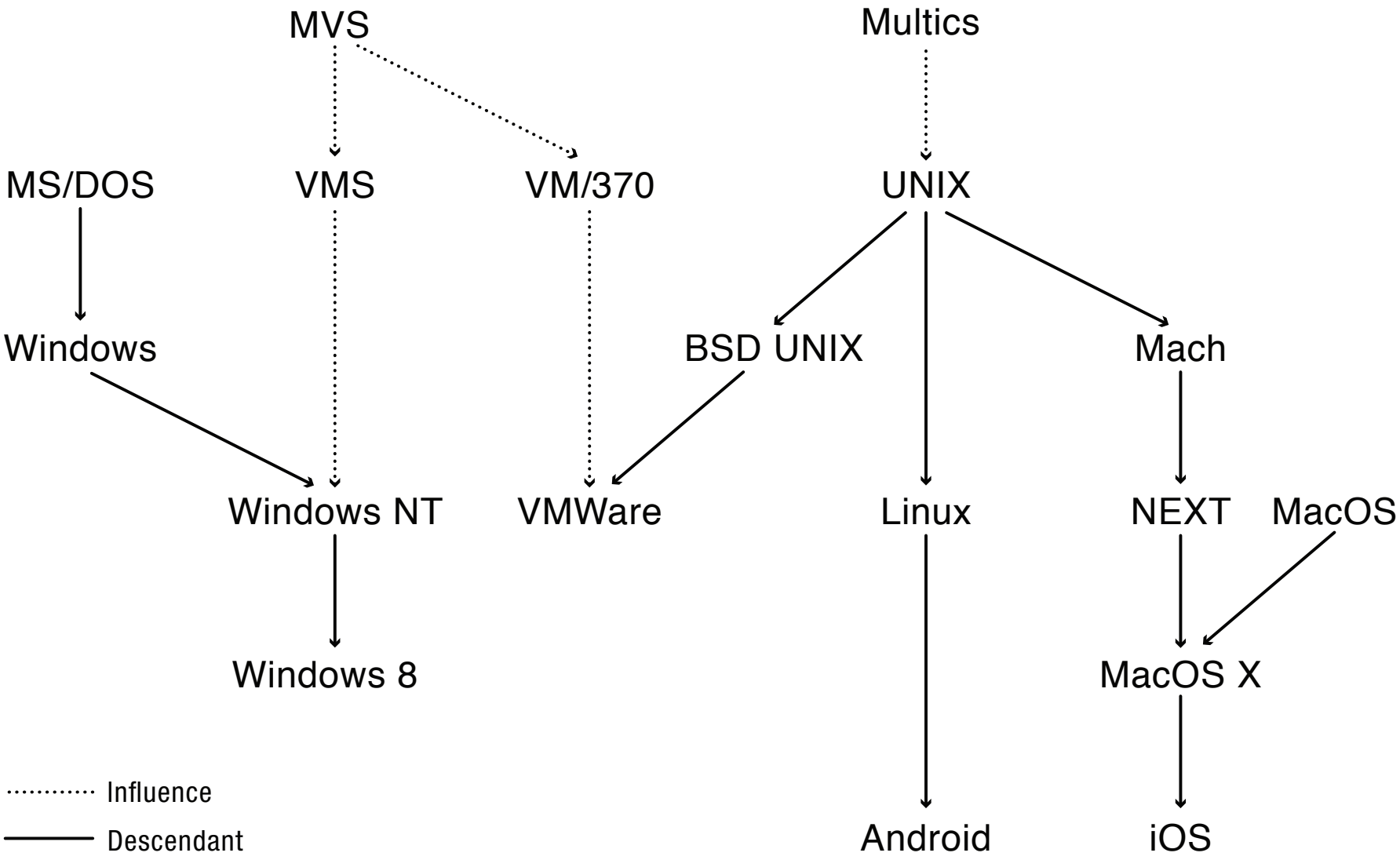


OS Challenges

Performance

- Latency/response time
  - How long does an operation take to complete?
- Throughput
  - How many operations can be done per unit of time?
- Overhead
  - How much extra work is done by the OS?
- Fairness
  - How equal is the performance received by different users?
- Predictability
  - How consistent is the performance over time?

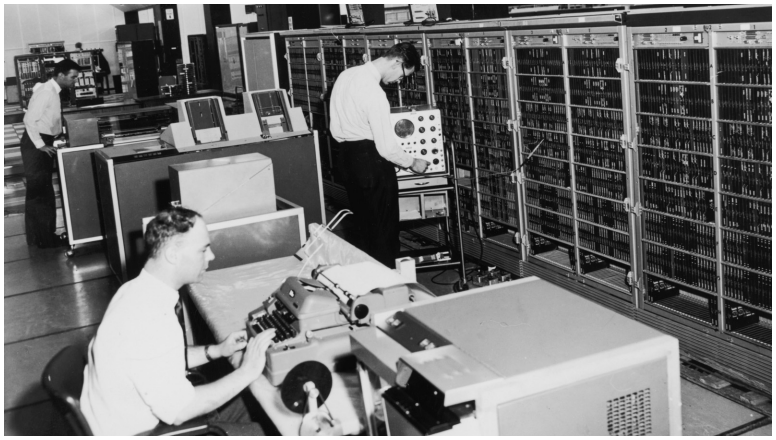
OS Family Tree



Performance / 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



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 OSs

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 OSs

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

Tomorrow's OSs

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