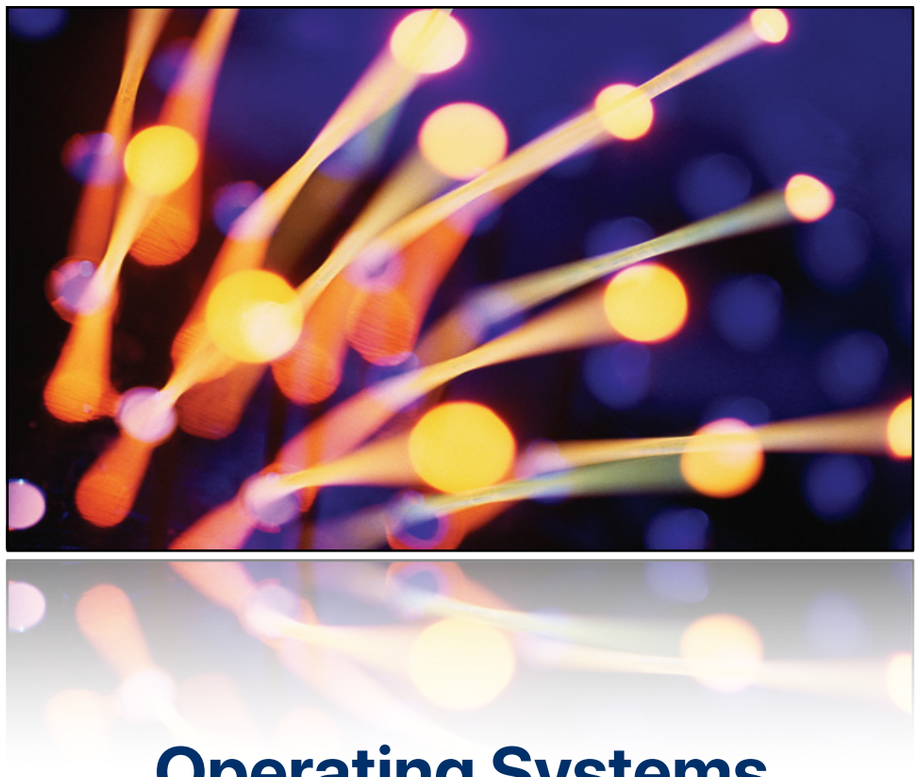
OS: a bird's-eye view

Part 1



Operating Systems

Baochun Li
University of Toronto

Last time

A brief history of operating systems

Baochun Li, Department of Electrical and Computer Engineering, University of Toronto

Last time

Operating system: a layer of software between applications and hardware

Baochun Li, Department of Electrical and Computer Engineering, University of Toronto

Last time

The OS provides an API to the applications above, and manages shared resources

Reading

Required: Three Easy Pieces

Chapter 2: Introduction to Operating Systems

What is an operating system?

The application developer's (or user's) view: "top-down"

OS designed to provide an Application Programming Interface (API) to make using hardware resources easier (called *system calls*, or *syscalls*)

Hides details via good abstractions

The system's view: resource manager ("bottom-up")

OS manages possibly conflicting requests for resources, such as CPU cycles, memory, and storage

It virtualizes physical resources

OS as an API (a standard library)

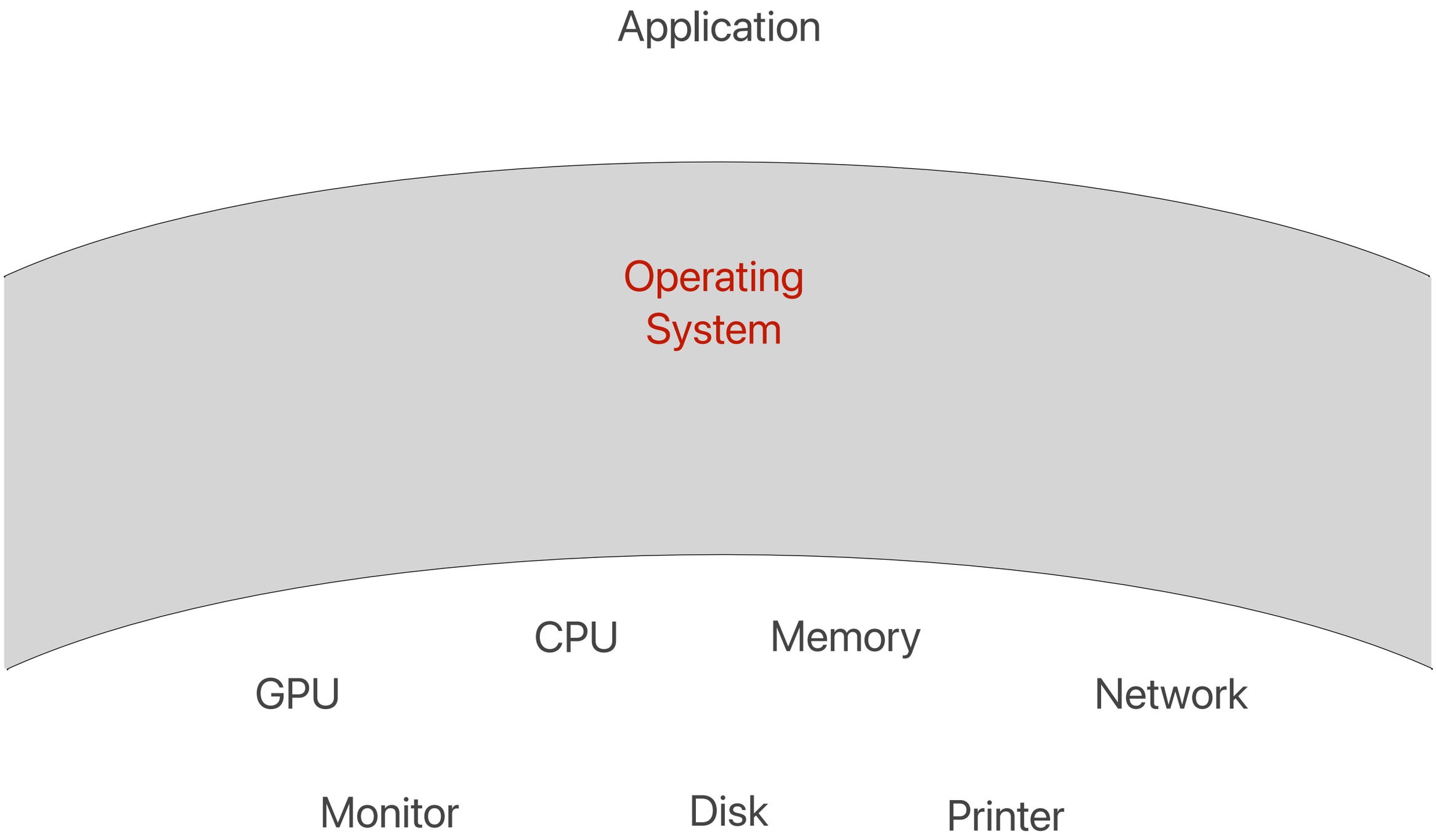
Why is such an abstraction important?

Otherwise, application writers must program all device accesses directly

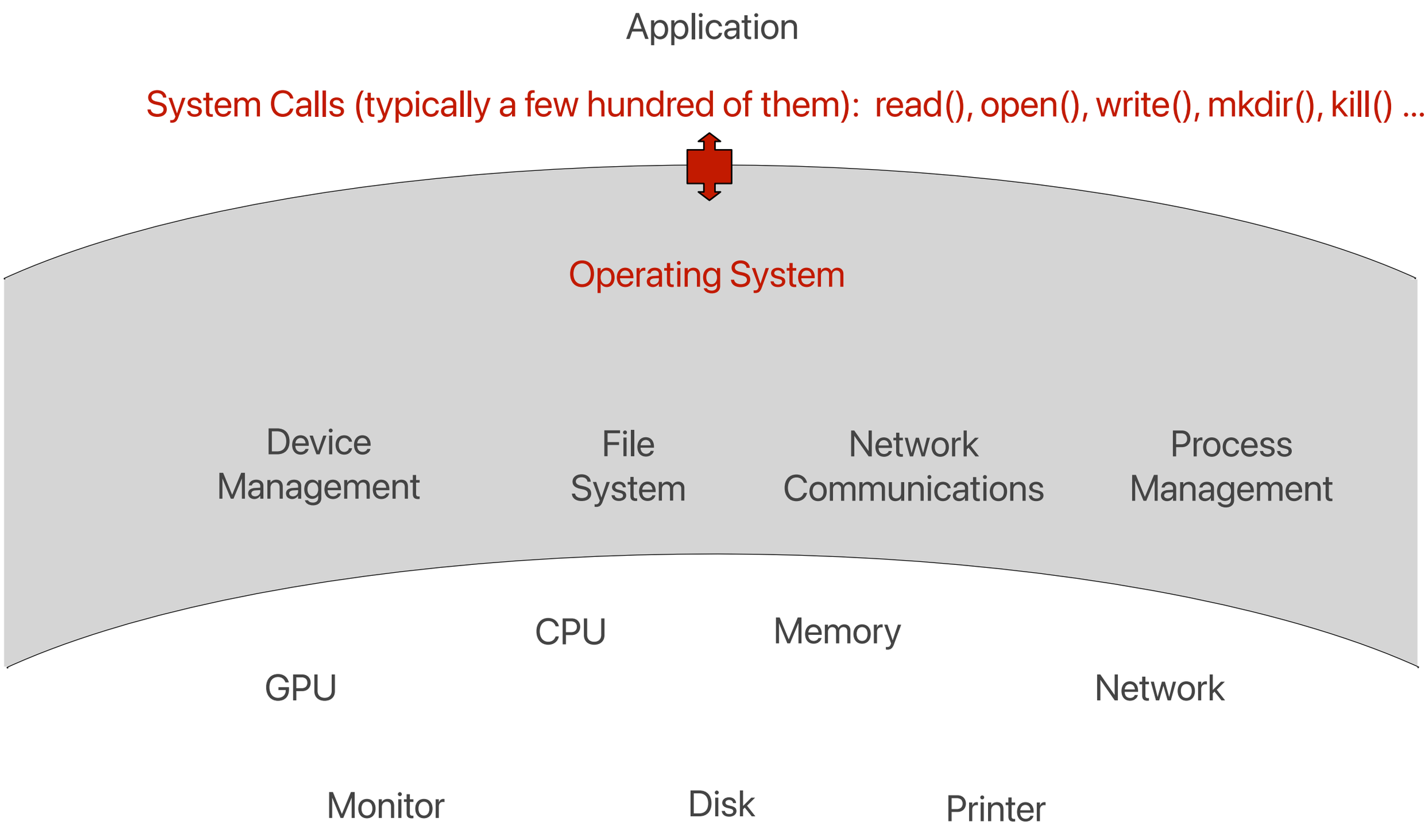
- Load device command codes into device registers
- Handle initialization and timing for physical devices
- Interpret return codes

Hard to maintain and upgrade code

Providing an API via system calls



Providing an API via system calls



OS as a resource manager

Shares resources across applications

- Sharing a resource (CPU) over time
- Sharing a resource (disk, memory) over space

Makes efficient use of a limited resource

- Improves utilization and performance
- Minimizes overhead

Protects applications from each other

- Enforces boundaries

The OS virtualizes resources — the OS takes a physical resource (CPU, memory, or a disk) and makes it easier to use.

Three “easy” pieces (main themes) in this course: virtualization, concurrency, and persistence

Baochun Li, Department of Electrical and Computer Engineering, University of Toronto

Theme #1

Virtualization: The OS takes a physical resource (CPU and memory) and transforms it into an easy-to-use virtual form of itself

To the applications, the OS is a virtual machine

Baochun Li, Department of Electrical and Computer Engineering, University of Toronto

The big question: how does the OS virtualize resources? — mechanisms and policies

Baochun Li, Department of Electrical and Computer Engineering, University of Toronto

Demo: Virtualizing the CPU