Introduction of OS

Shengzhong Liu
Department of Computer Science and Engineering
Shanghai Jiao Tong University

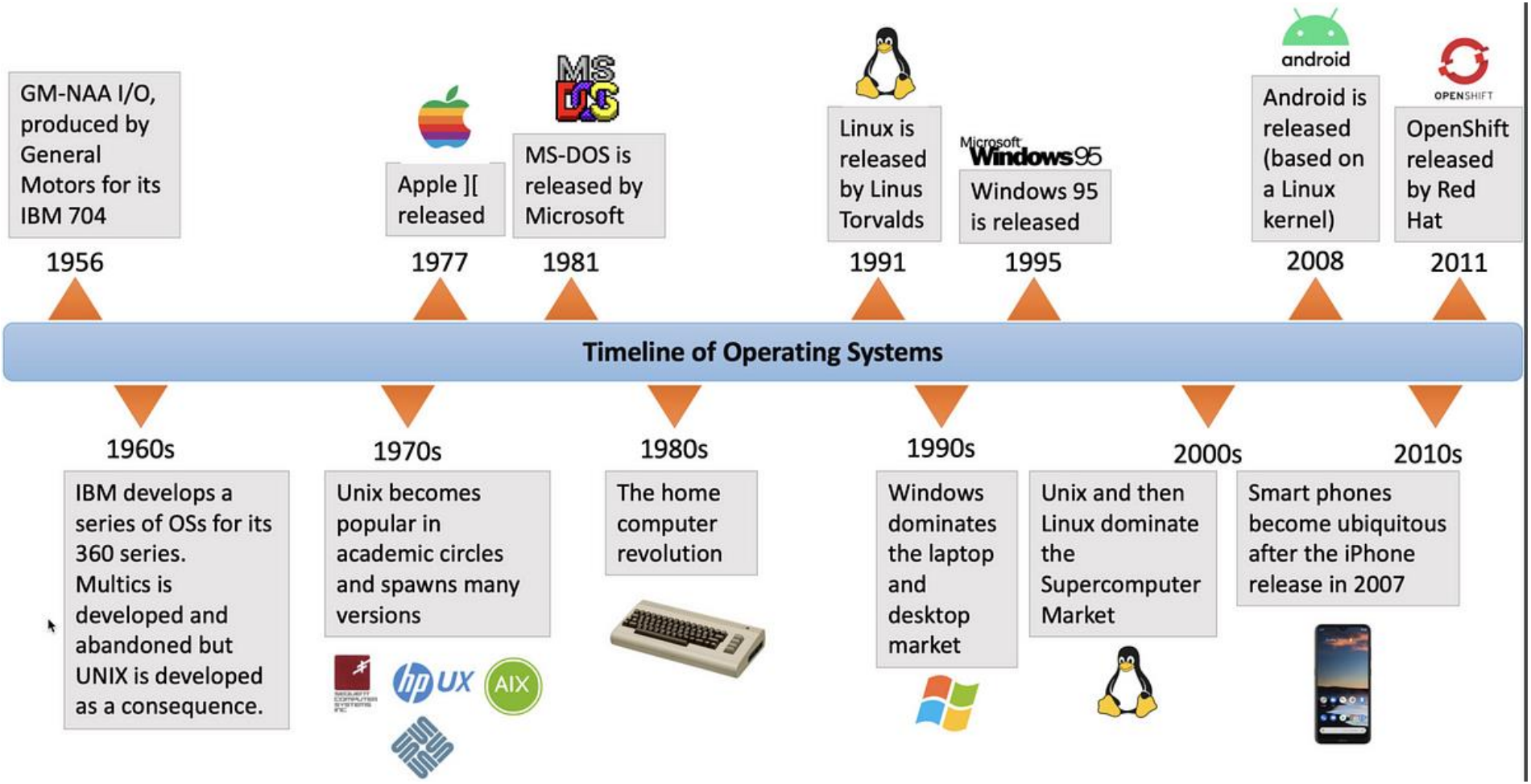
Part of slides from UIUC CS423 course.

Outline

- Operating System History
- Recap on Computer System Organization
- What Operating Systems Do
- Operating System Tasks
- Operating System Evolvement
- Computer System Environments

Operating System History

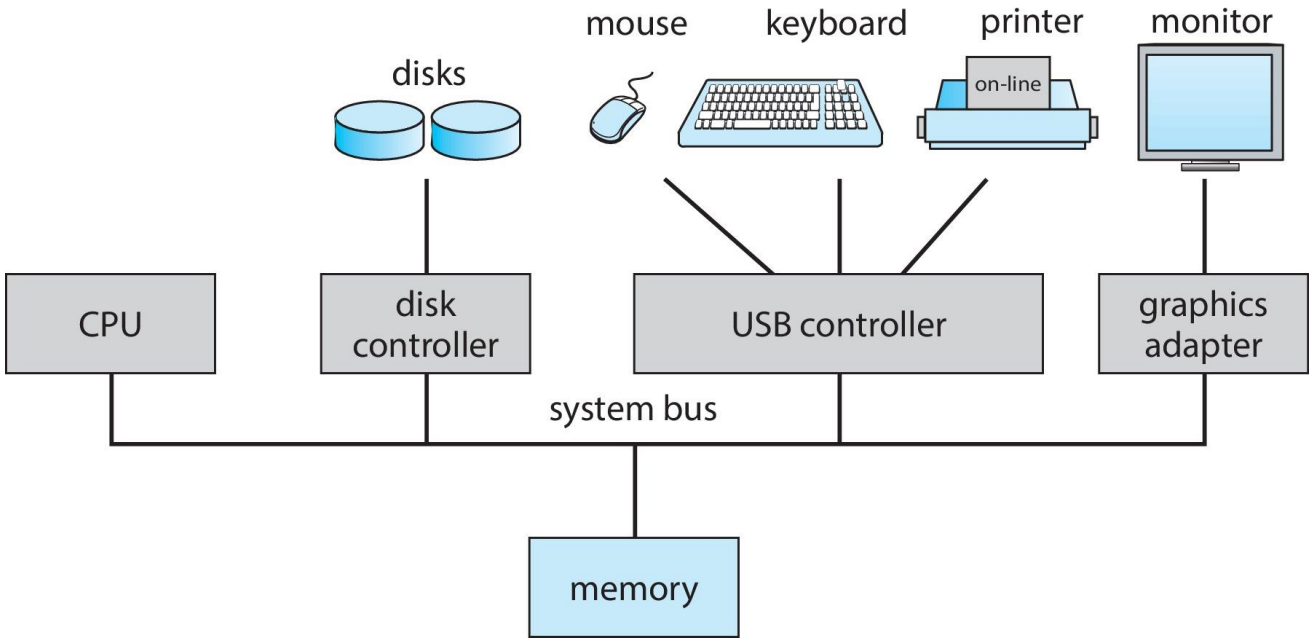
OS History


GM-NAA I/O, produced by General Motors for its IBM 704
1956
Apple][released
1977
MS-DOS is released by Microsoft
1981
Linux is released by Linus Torvalds
1991
Windows 95 is released
1995
Android is released (based on a Linux kernel)
2008
OpenShift released by Red Hat
2011
Timeline of Operating Systems
1960s
IBM develops a series of OSs for its 360 series. Multics is developed and abandoned but UNIX is developed as a consequence.
1970s
Unix becomes popular in academic circles and spawns many versions
1980s
The home computer revolution
1990s
Windows dominates the laptop and desktop market
2000s
Unix and then Linux dominate the Supercomputer Market
2010s
Smart phones become ubiquitous after the iPhone release in 2007

Recap on Computer System Organization

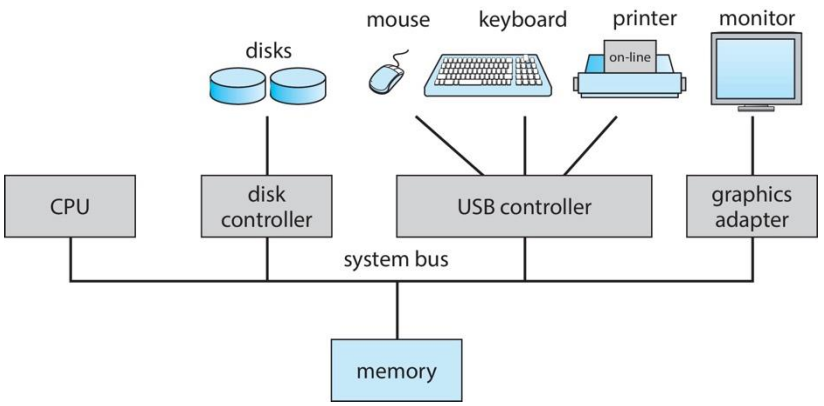
Computer System Organization

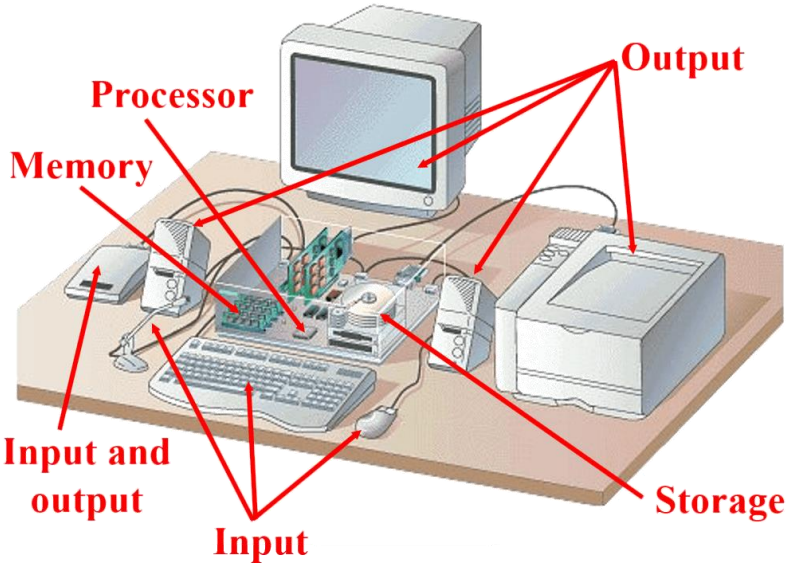
- Computer-system operation
  - One or more CPUs, device controllers connect through **common bus** providing access to shared memory
  - Concurrent execution of CPUs and devices competing for memory cycles



Computer-System Operation

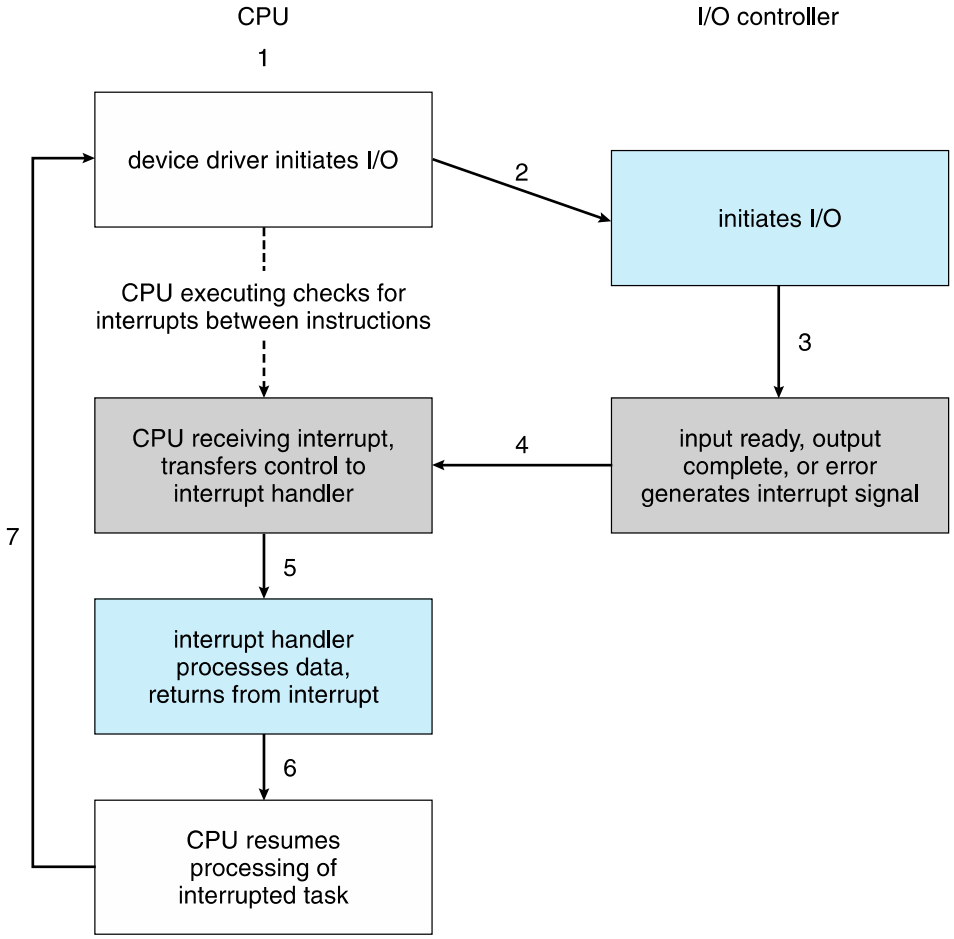
- Each device controller
  - In charge of a particular device type
  - Has a local buffer
  - Each type has an operating system **device driver（设备驱动）**to manage it
- I/O devices and the CPU can execute concurrently
  - CPU moves data between main memory and controller local buffers
  - I/O moves data between device and controller local buffers
- Device controller informs CPU that it has finished its operation by causing an **interrupt (中断)**





Common Functions of Interrupts

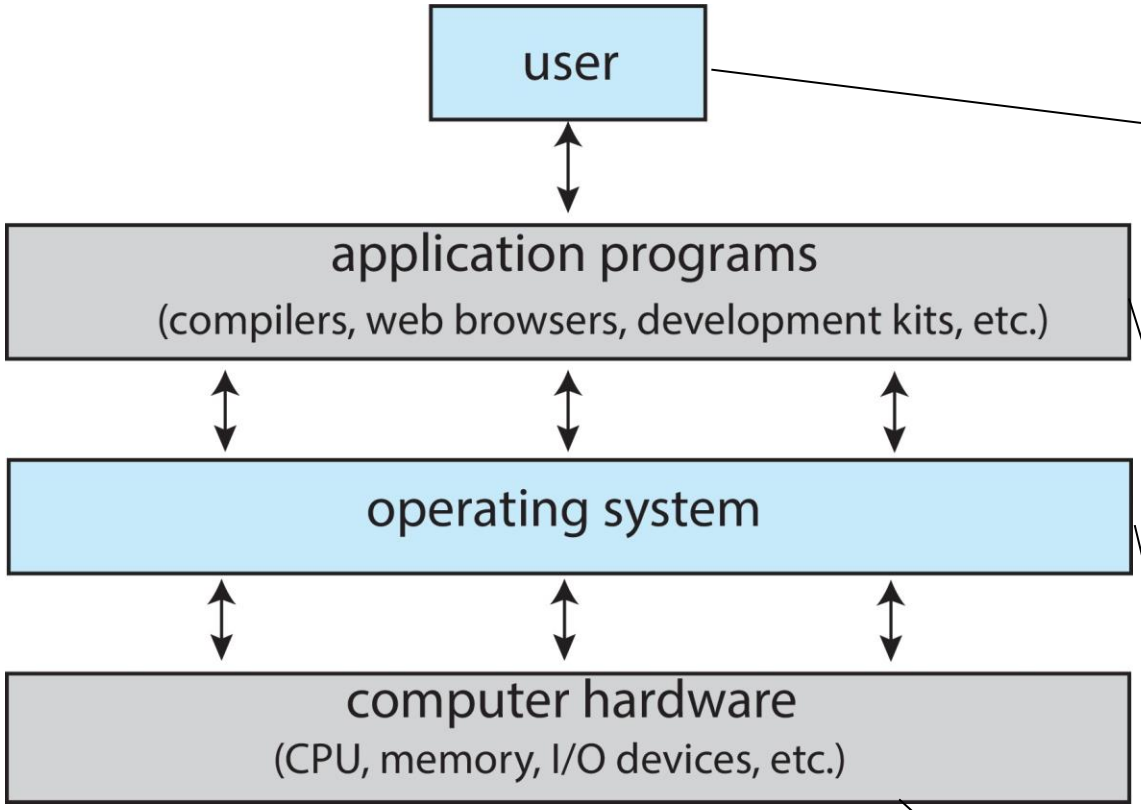
- Interrupt transfers control to the interrupt service routine generally, through the **interrupt vector**
  - The interrupt vector contains the addresses of all the service routines
- Interrupt architecture must save the address of the interrupted instruction
- An operating system is **interrupt driven**
  - A **trap** or **exception** is a software-generated interrupt caused either by an error or a user request



Interrupt-driven I/O cycle

What Operating Systems Do?

Abstract View of Computer Components



People, machines, other computers

Application programs define the ways in which the system resources are used to solve the computing problems of

Controls and coordinates use of hardware among various applications and users

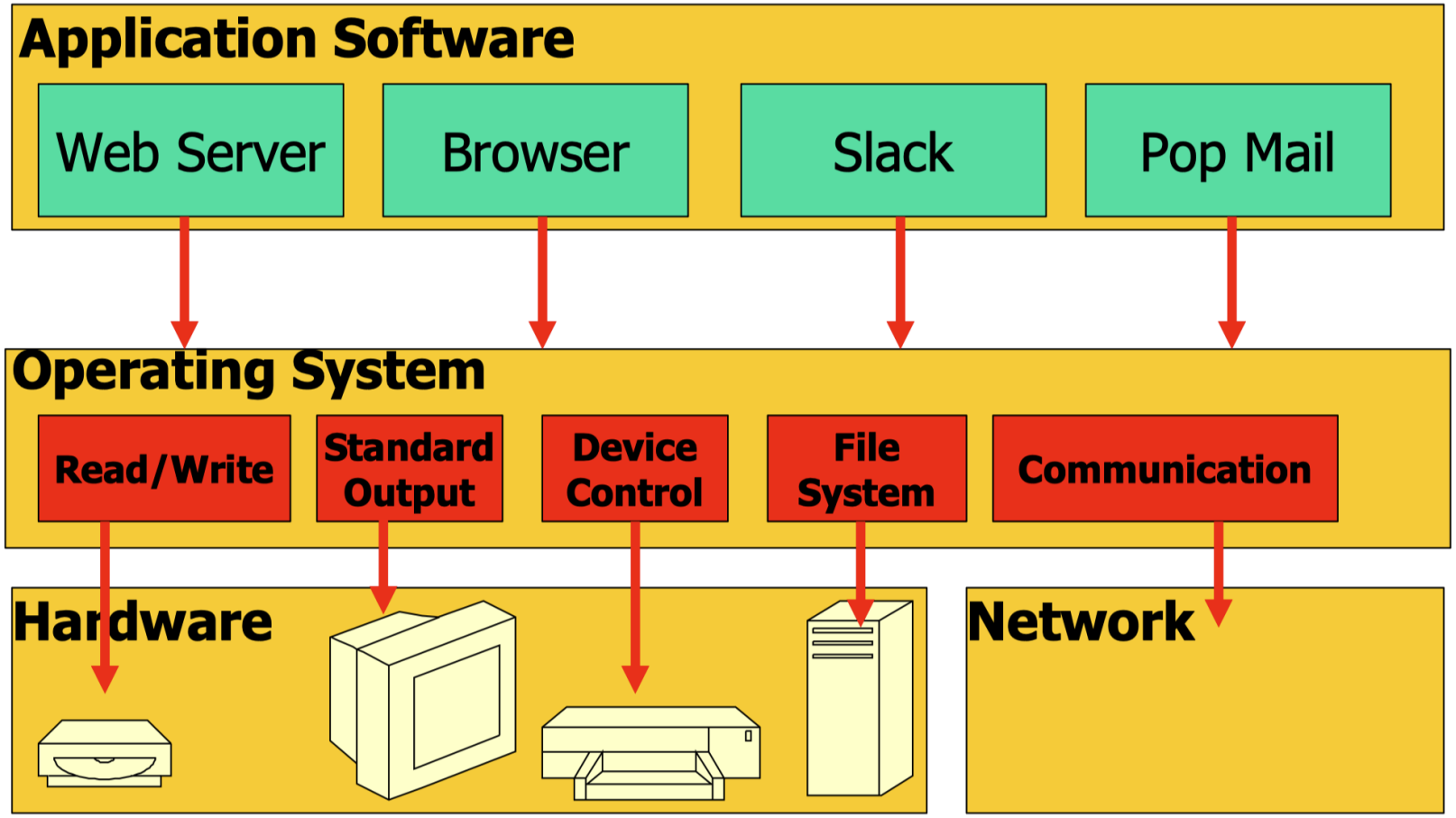
provides basic computing resources

Computer System Structure

- Computer system can be divided into four components:
  - **Users**
    - People, machines, other computers
  - **Application programs** – define the ways in which the system resources are used to solve the computing problems of the users
    - Word processors, compilers, web browsers, database systems, video games
  - **Operating system** – Controls and coordinates use of hardware among various applications and users
  - **Hardware** – provides basic computing resources
    - CPU, memory, I/O devices

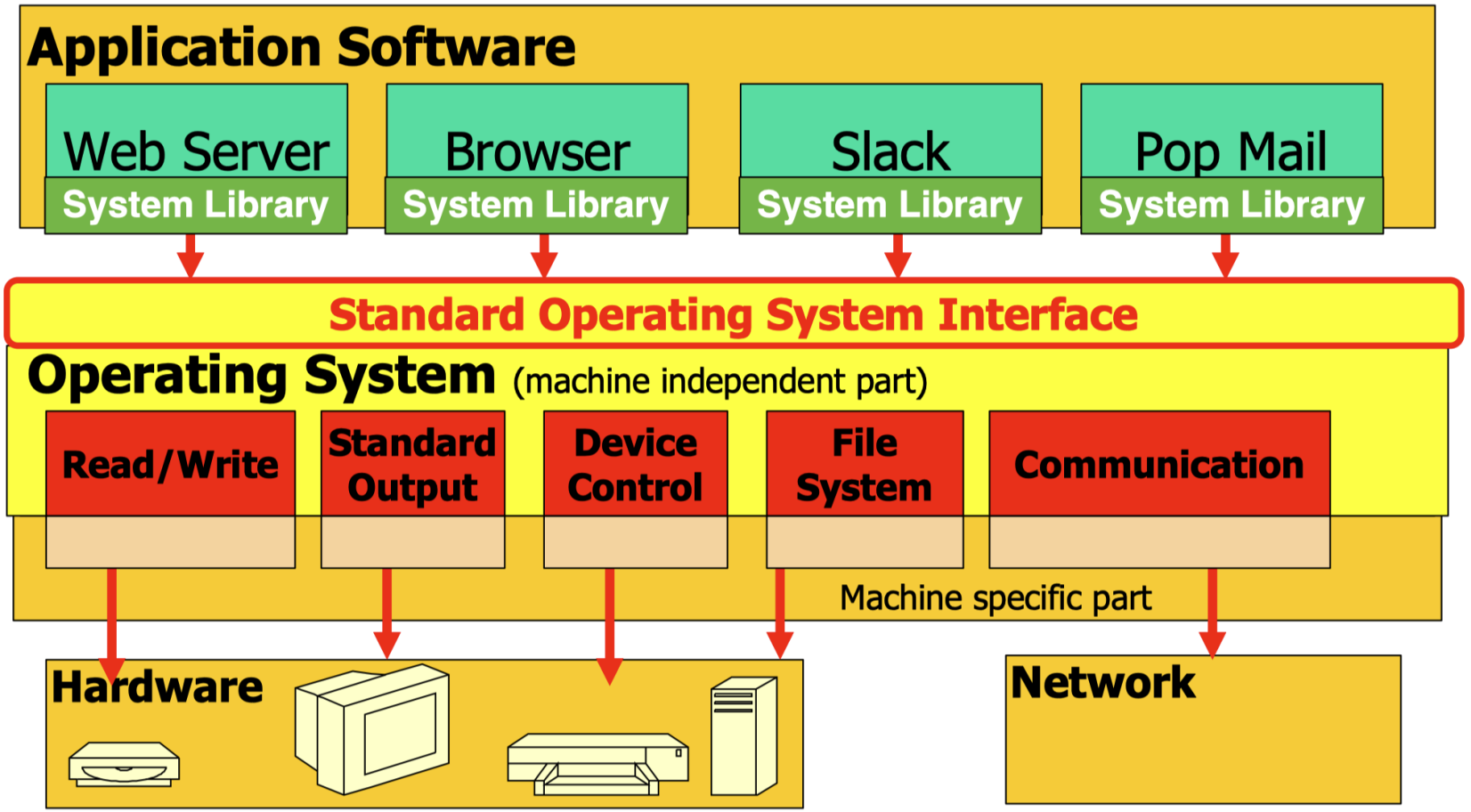
Why Operating Systems?

Software to manage the computer resources for its users



Why Operating Systems?

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



What is an Operating System?

- An operating system is a **program** that manages the computer hardware
- An **intermediary** between **computer user** and **computer hardware**
  - Users want convenience, ease of use, and good performance.
  - Shared computers must keep all users happy
- Operating system goals:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

Operating System Definition

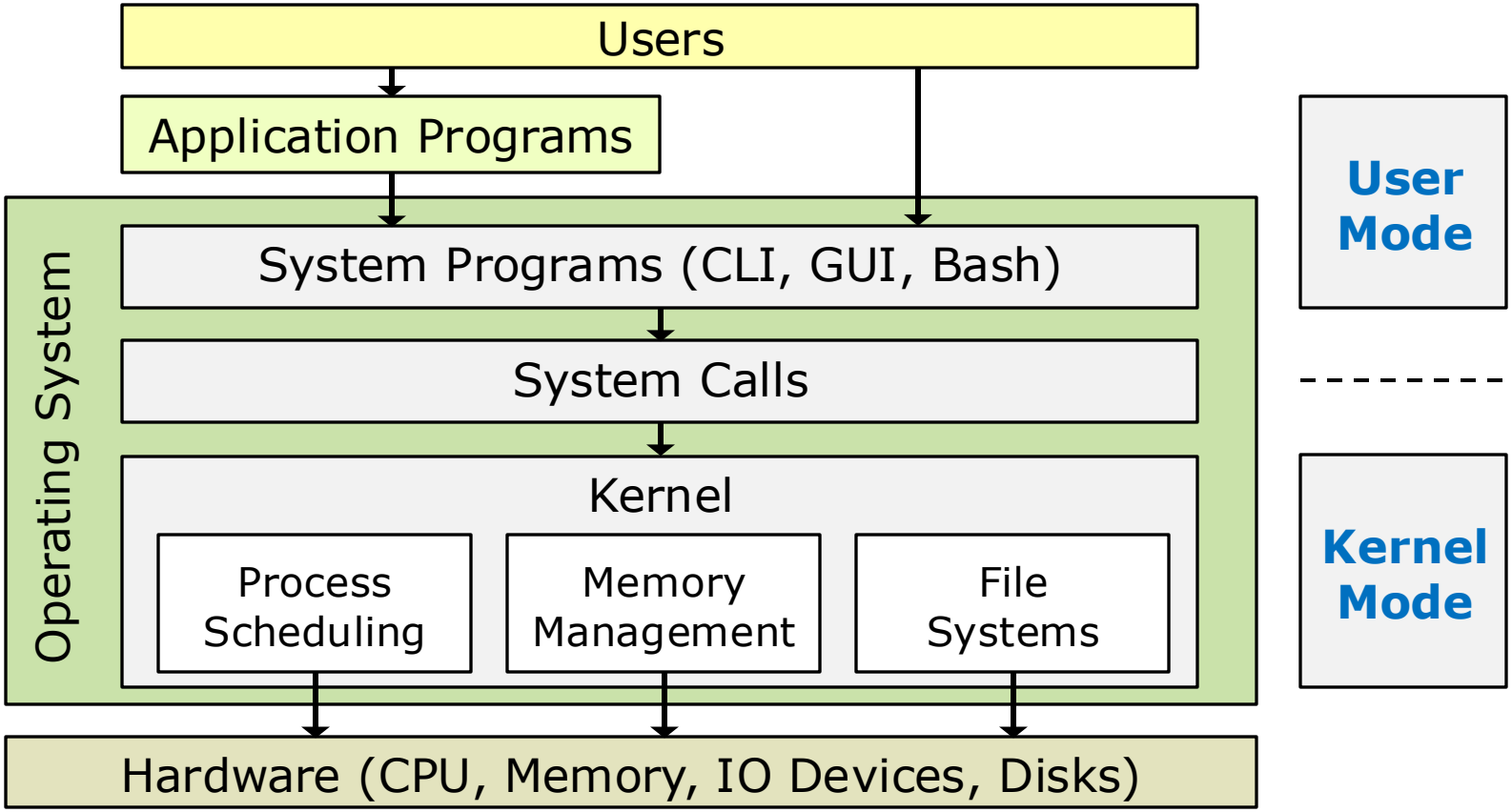
- Term OS covers many roles
  - Because of myriad designs and uses of OSes
  - Present in toasters through ships, spacecraft, game machines, TVs and industrial control systems

- OS is a **resource allocator**
  - Manages all underlying hardware resources
  - Decides between conflicting requests for efficient and fair resource use

- OS is a **control program**
  - Controls execution of programs to prevent errors and improper use of the computer

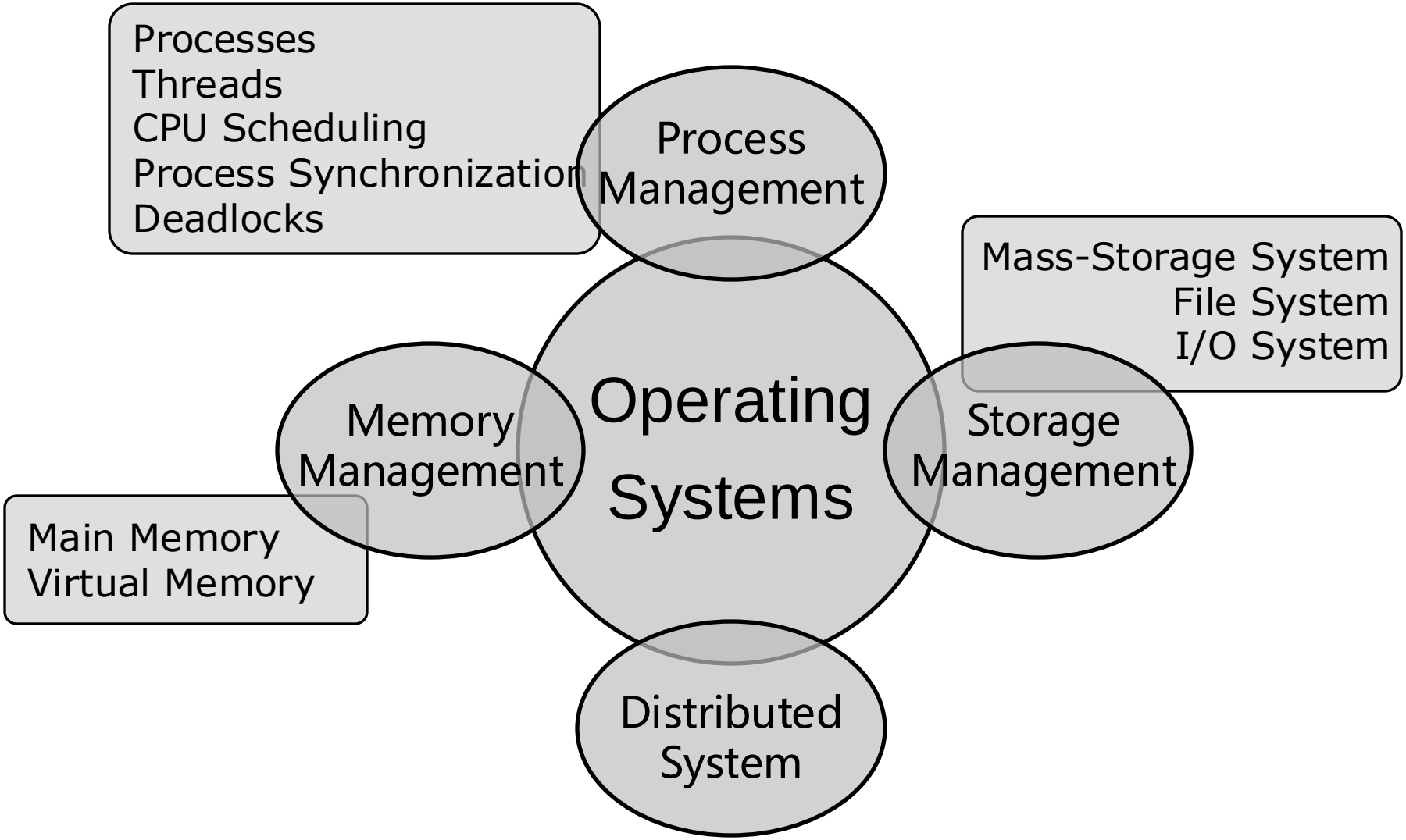
Operating System Definition (Cont.)

- Approximated definition of OS: “Everything a vendor ships when you order an operating system”
- “The one program running at all times on the computer” is the **kernel**, part of the operating system.



Operating System Tasks

Operating System Tasks



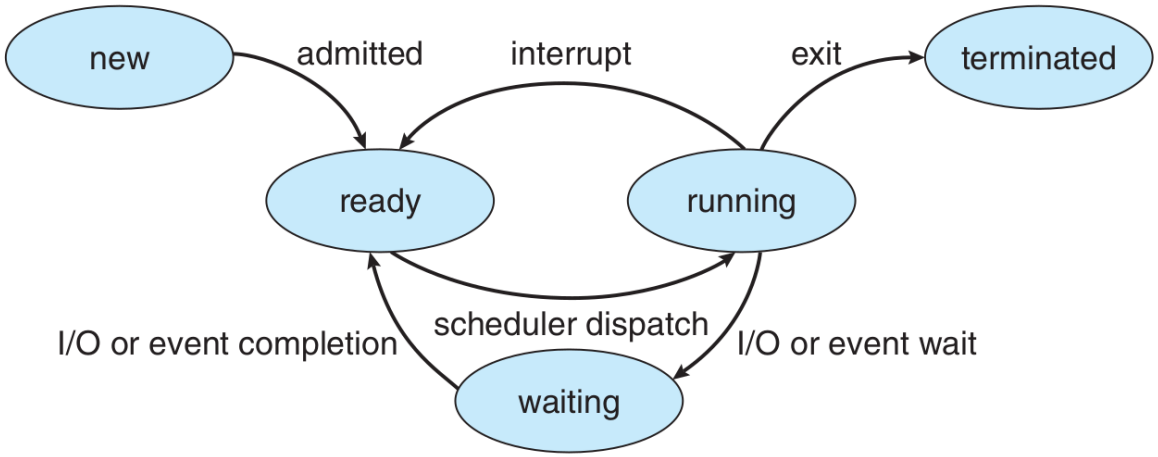
Task 1: Process Management

- A process is a program in execution. It is a unit of work within the system.
  - Program is a ***passive entity;*** process is an ***active entity***.
- Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Initialization data
- Process executes instructions sequentially, one at a time, until completion
  - Single-threaded process has one **program counter**
  - Multi-threaded process has **one program counter per thread**
- Typical system has many processes running concurrently on one or more CPUs

Process Management Activities

The operating system is responsible for the following activities in connection with process management:

- Creating and deleting both user and system processes
- Suspending and resuming processes
- Providing mechanisms for process synchronization and communication
- Providing mechanisms for deadlock handling

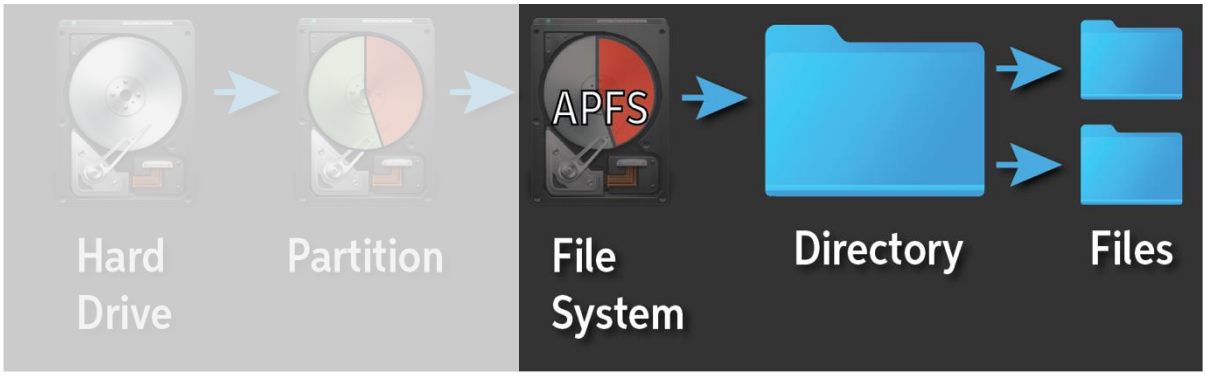


Task 2: Memory Management

- To execute a program
  - All (or part) of the instructions must be in memory
  - All (or part) of the data that is needed by the program must be in memory
- **Memory management determines what is in memory and when**
  - Optimizing CPU utilization and computer response to users
- Memory management activities
  - Track which parts of memory are currently being used and by whom
  - Decide which processes and data to move into and out of memory
  - Allocate and deallocate memory space as needed

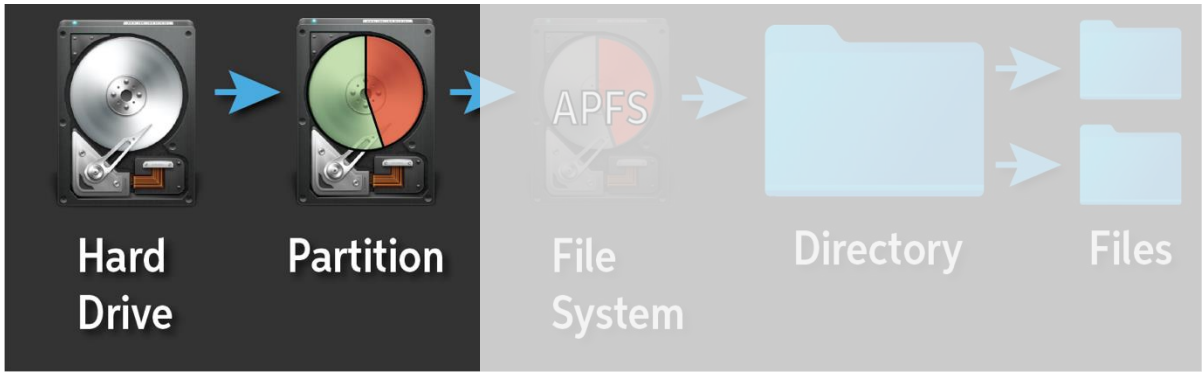
Task 3: File-system Management

- OS provides uniform, logical view of information storage
  - **File**: Abstracts physical properties to logical storage unit
- File-System management
  - Files usually organized into directories
  - Access control on most systems to determine who can access what
- File-system management activities:
  - Creating and deleting files and directories
  - Primitives to manipulate files and directories
  - Mapping files onto secondary storage
  - Backup files onto stable (non-volatile) storage media



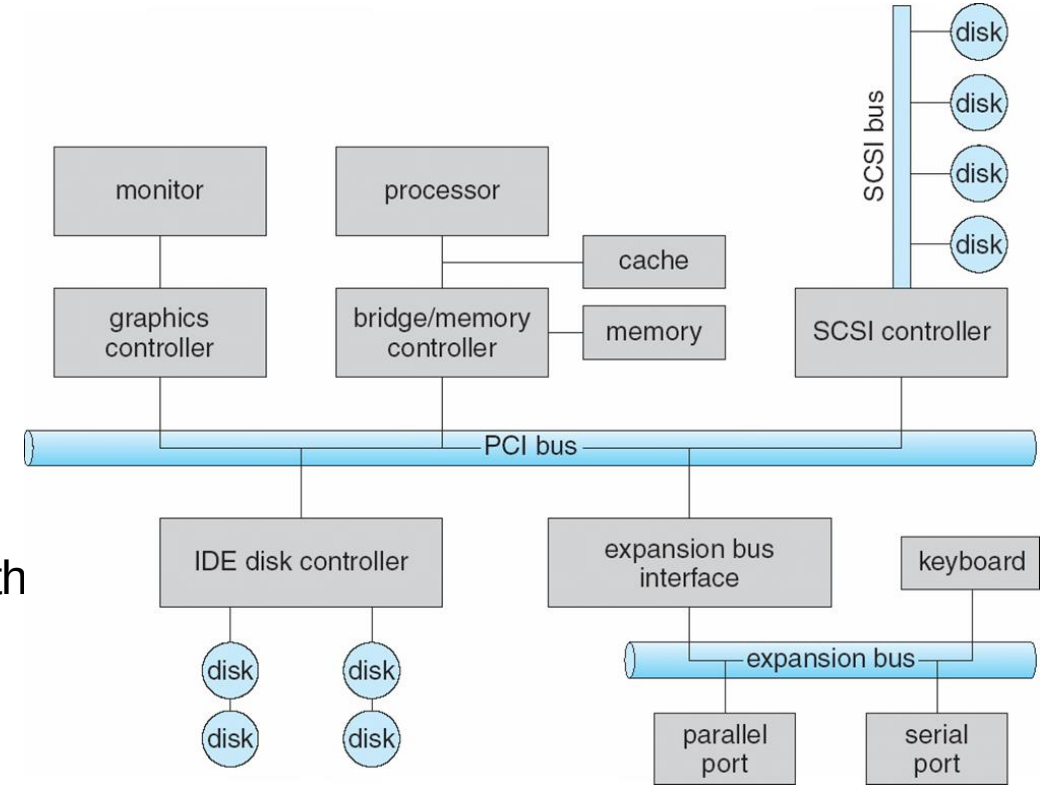
Task 4: Mass-Storage Management

- Disks are used to store data that does not fit in main memory or data that must be kept for a “long” period of time
- The entire speed of computer operation depends on the disk and its algorithms
- Mass-storage management activities
  - Mounting and unmounting
  - Free-space management
  - Storage allocation
  - Disk scheduling
  - Partitioning
  - Protection



Task 5: I/O Systems

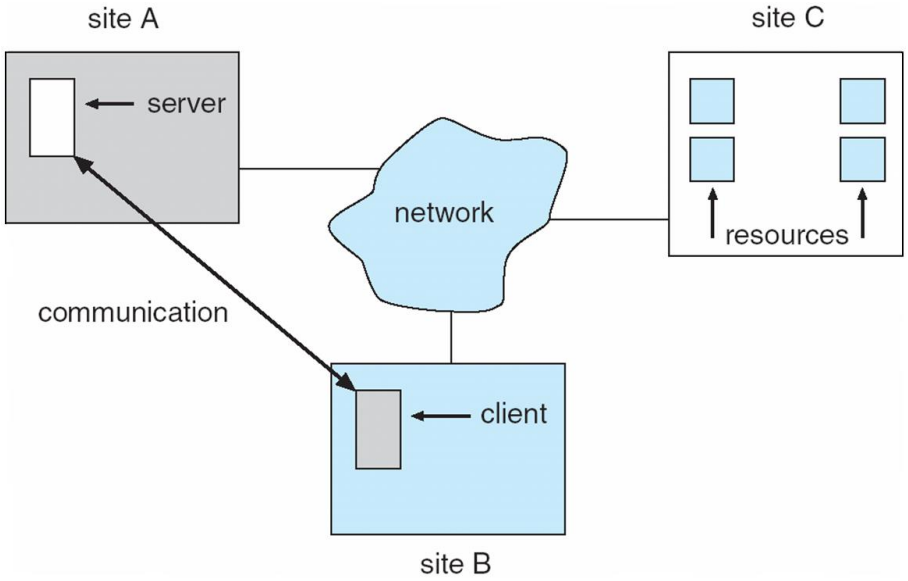
- One purpose of OS is to hide the features of hardware devices from the user
- I/O system responsible for:
  - I/O memory management:
    - **Buffering**: storing data temporarily while transferring
    - **Caching**: storing parts of data in faster storage for performance
    - **Spooling**: the overlapping of the output of one job with the input of other jobs
  - Device-driver interface
  - Drivers for specific hardware devices



Extension: Distributed Systems

- Collection of separate, possibly heterogeneous, systems networked together

- **Network Operating System** provides features between systems across network
  - Communication scheme allows systems to exchange messages
  - Illusion of a single system



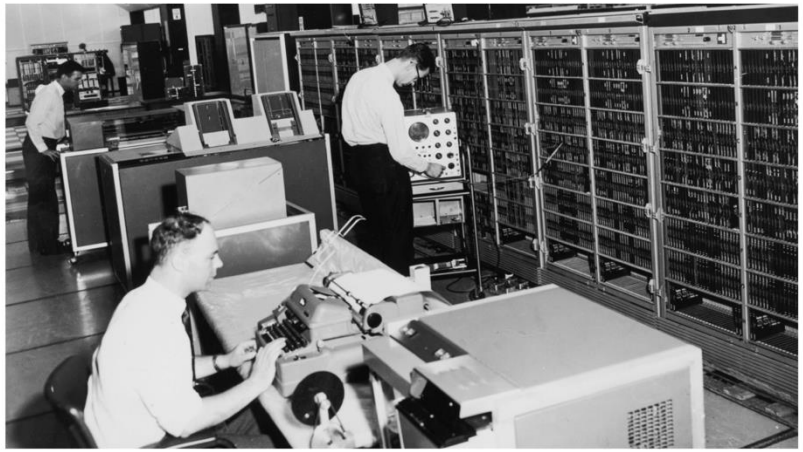
Operating System Evolvement

Operating System Evolvement

	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+

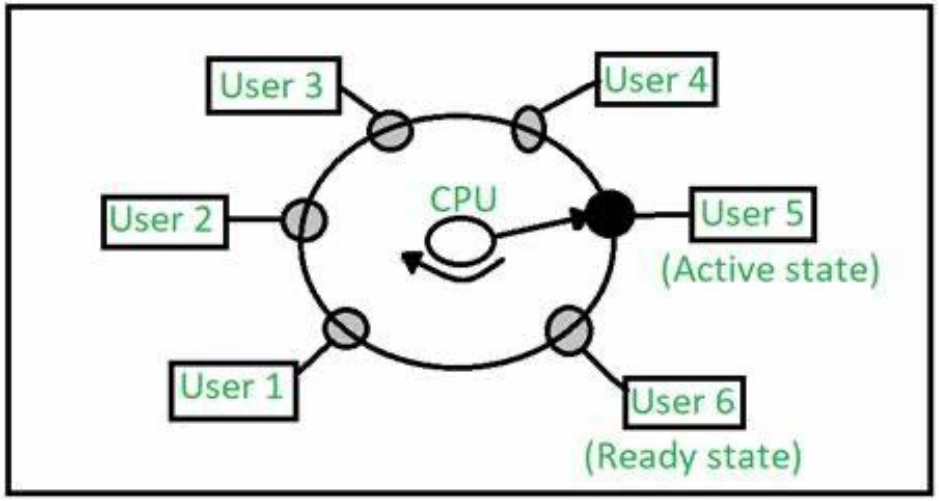
Early Operating Systems

- One application at a time:
  - Had complete control of hardware
  - OS was runtime library
  - Users would stand in queues 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 OSes

- Multiple users on the computer at the same time
  - Multiprogramming: run multiple programs at the 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

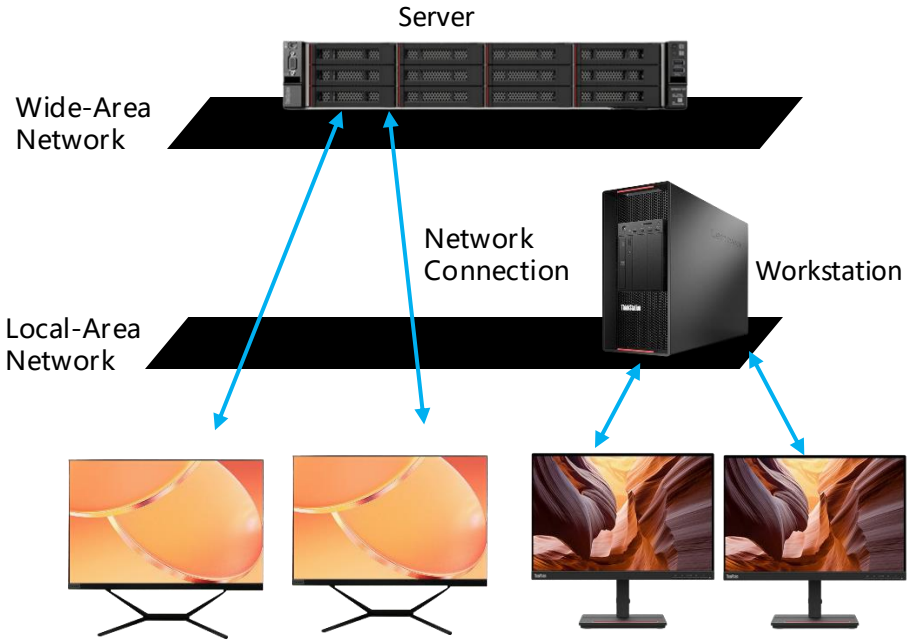


Today's OSes

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

Tomorrow's OSes

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



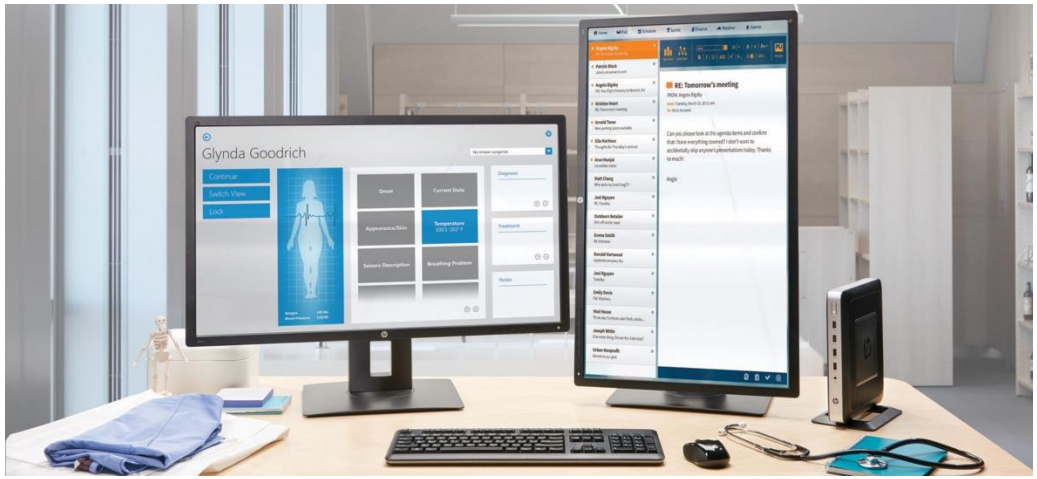
Computer System Paradigms

Computing Paradigms

- Traditional
- Mobile
- Client Server
- Peer-to-Peer
- Cloud computing
- Real-time Embedded
- Intelligent Computing: LLM + OS

Traditional

- Stand-alone general-purpose machines
  - Blurred as most systems interconnect with others (i.e., the Internet)
- **Portals** provide web access to internal systems
- **Network computers** (**thin clients 瘦客户机**) are like Web terminals
- Mobile computers interconnect via **wireless networks**
- Networking becoming ubiquitous – even home systems use **firewalls** to protect home computers from Internet attacks

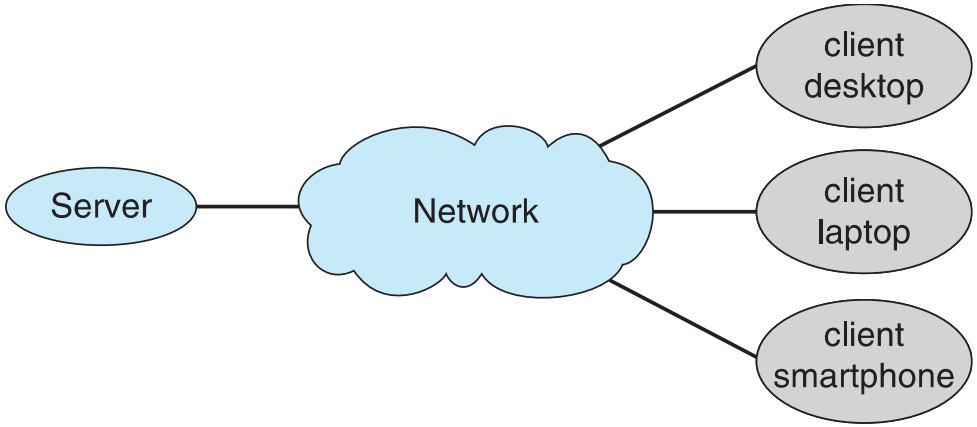


Mobile

- Handheld smartphones, tablets, etc.
- Extra feature – more OS features (GPS, gyroscope)
- Allows new types of apps like ***augmented reality***
- Use IEEE 802.11 wireless, or cellular data networks for connectivity
- Leaders are **Apple iOS, Google Android, Huawei Harmony OS**

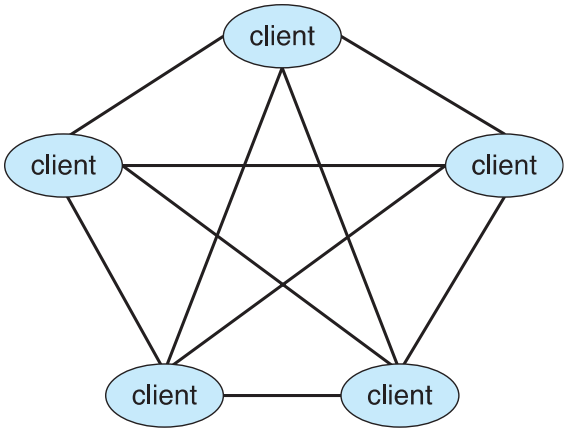
Client Server

- Client-Server Computing
  - Dumb terminals supplanted by smart PCs
  - Many systems now **servers**, responding to requests generated by **clients**
    - **Compute-server system** provides an interface to client to request services (i.e., database)
    - **File-server system** provides interface for clients to store and retrieve files



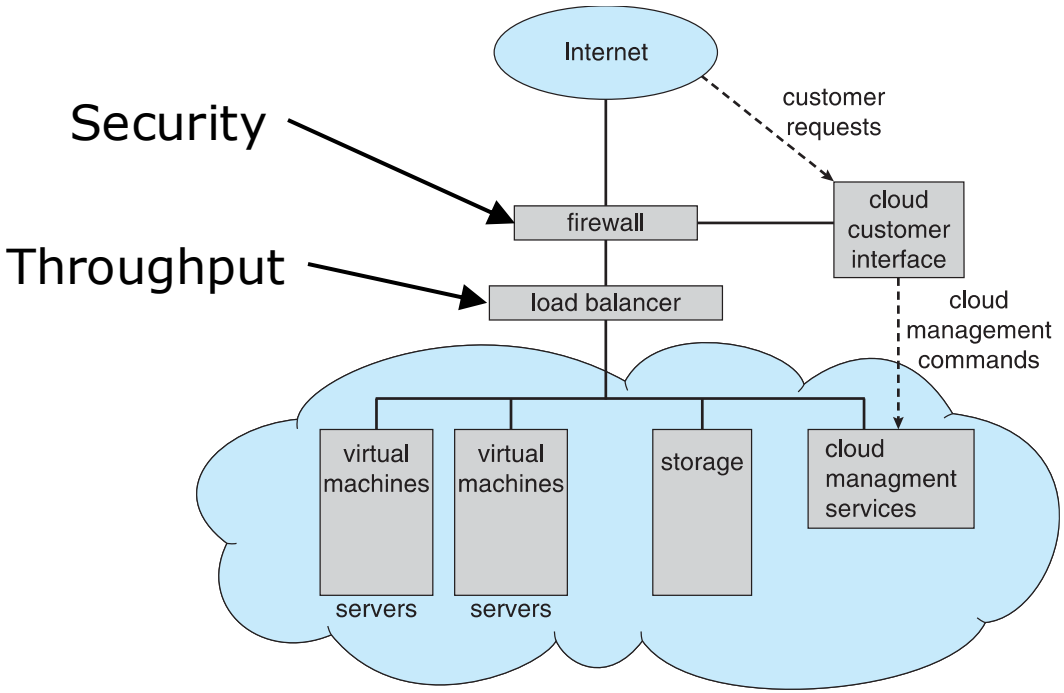
Peer-to-Peer

- Another model of distributed system
- P2P does not distinguish clients and servers
  - Instead all nodes are considered peers
  - May each act as client, server or both
  - Node must join P2P network
    - Registers its service with central lookup service on network, or
    - Broadcast request for service and respond to requests for service via ***discovery protocol***
  - Examples include Napster and Gnutella, **Voice over IP** (**VoIP**) such as Skype



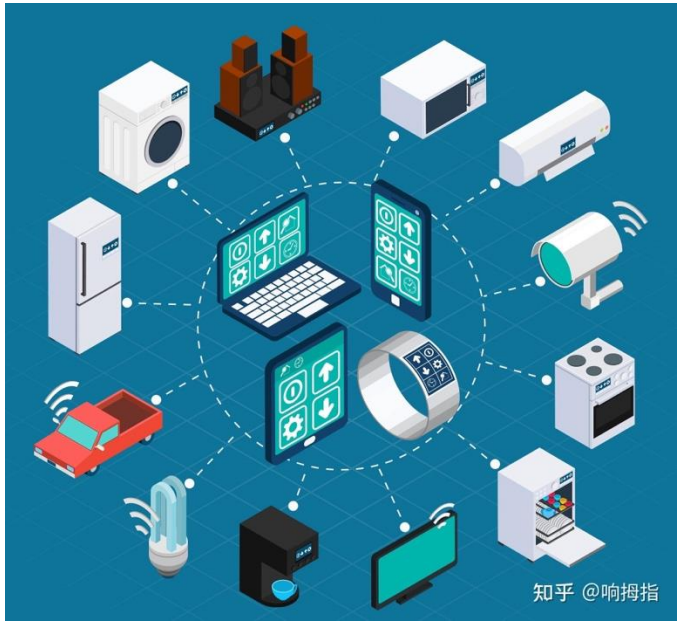
Cloud Computing

- Delivers *computing*, *storage*, even *apps as a service* across a network
- Logical extension of virtualization because it uses virtualization as the base for its functionality.



Real-Time Embedded Systems

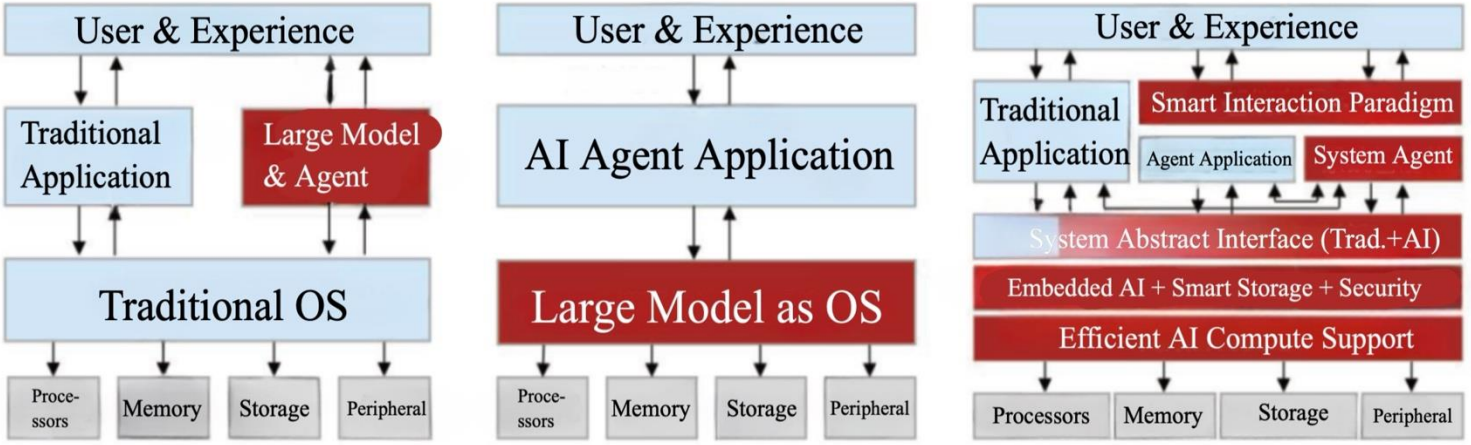
- Real-time embedded systems most prevalent form of computers
  - Vary considerable, special purpose, limited purpose OS, **real-time OS**
  - Use expanding
- Many other special computing environments as well
  - Some have OSes, some perform tasks without an OS
- Real-time OS has well-defined fixed time constraints
  - Processing ***must*** be done within constraint
  - Correct operation only if constraints met



Intelligent Computing: LLM + OS

- How to use LLM to enable the intelligence of OS is currently under active exploration
  - Everyone has their own Javis

- Three candidate approaches for LLM + OS:
  - LLM on top of OS
  - LLM as OS
  - LLM combined with OS



Homework

- Reading
  - Chapter 1: Introduction