Lecture 1
Introduction to Operating Systems

Prof. Yinqian Zhang

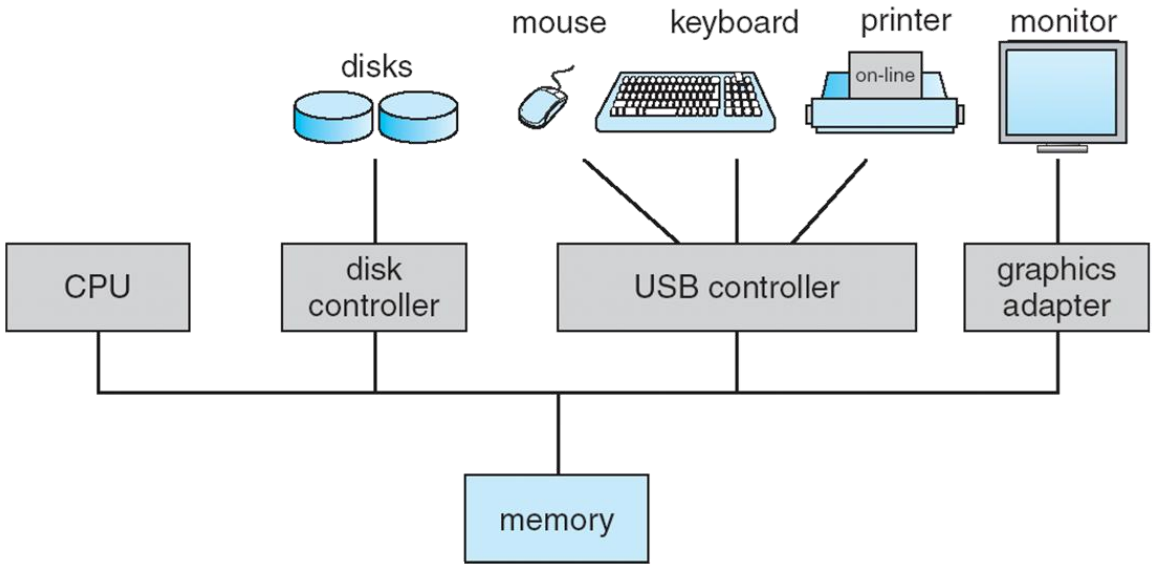
Fall 2025

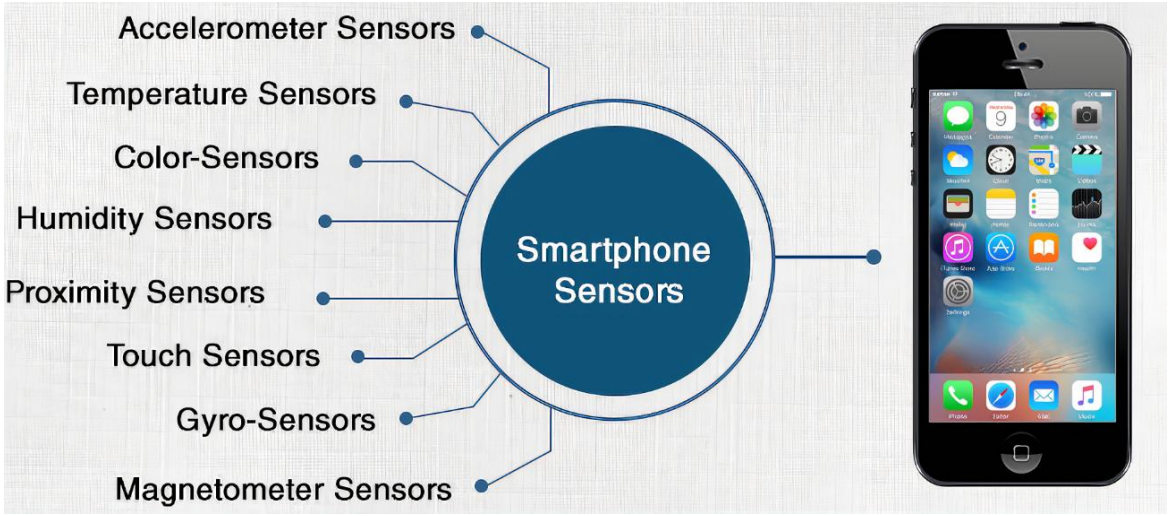
What is a Computer?

- A computer is a digital electronic machine that can be programmed to carry out sequences of arithmetic or logical operations automatically.

Computer System Organization

- 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

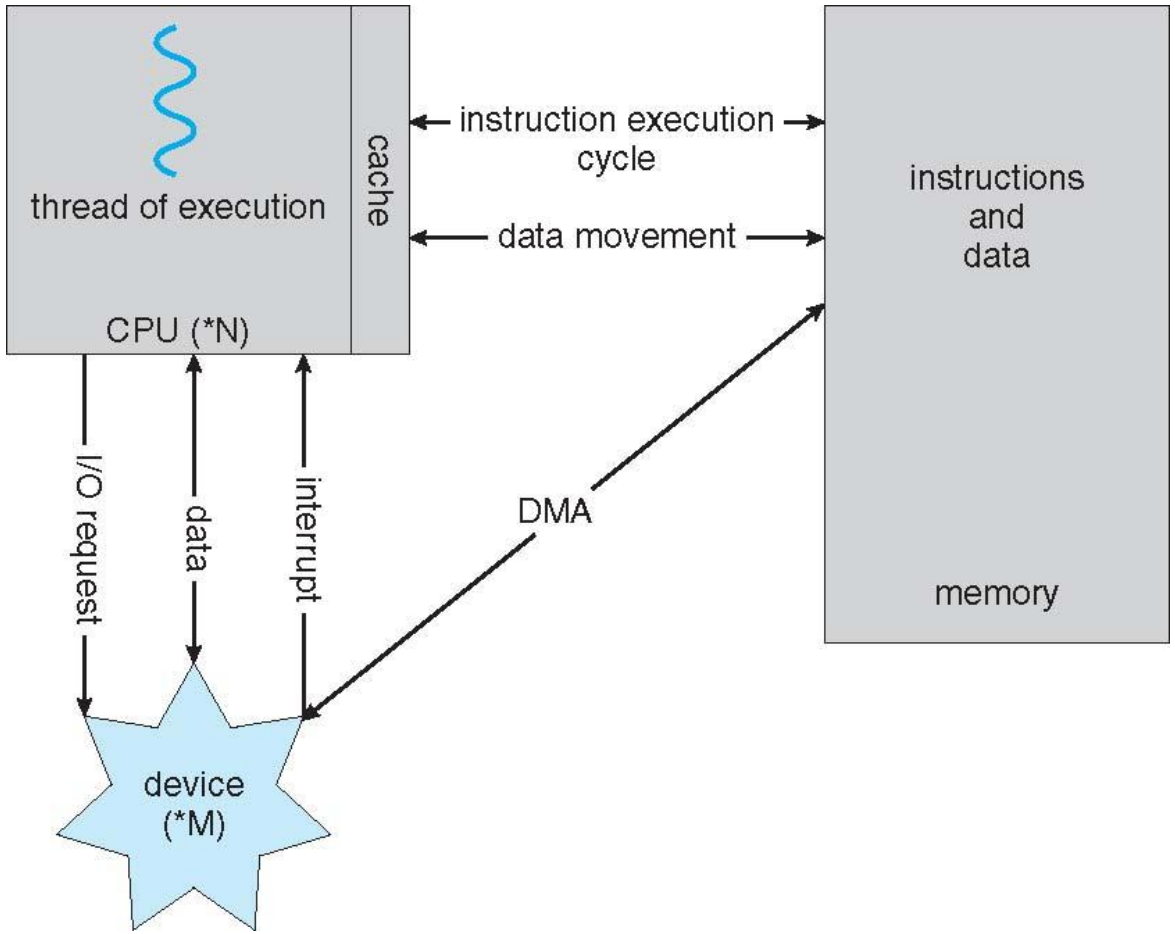




How a Modern Computer Works

- von Neumann Architecture:
  - a single, shared memory for programs and data
  - a single bus for memory access
  - an arithmetic unit
  - and a program control unit

CPU 和内存之间通过同一条通道传输指令和数据，因此在某些时候会有“瓶颈”（称为 冯·诺依曼瓶颈）



John von Neumann (1903~1957)

- Mathematician, computer scientist, physicist, chemist
- Known for invention of modern computer architecture and game theory

Structure of a Computer System

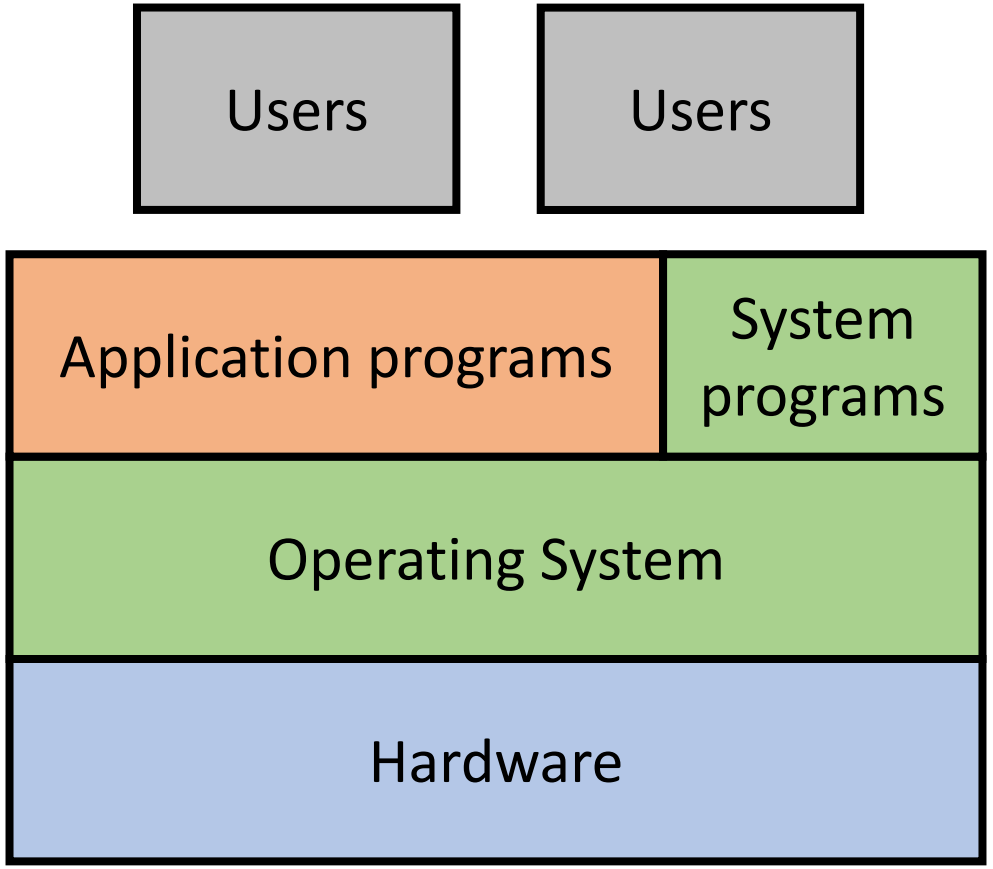
- Computer system can be divided into four components
  - Hardware
    - Provides basic computing resources
    - CPU, memory, I/O devices
  - Operating system
    - Controls and coordinates use of hardware among various applications and users
  - Application programs
    - Define the ways in which the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

控制并协调硬件资源的使用

定义了用户解决问题的方式

Structure of a Computer System

- Hardware
- Operating system
- Application programs
- Users

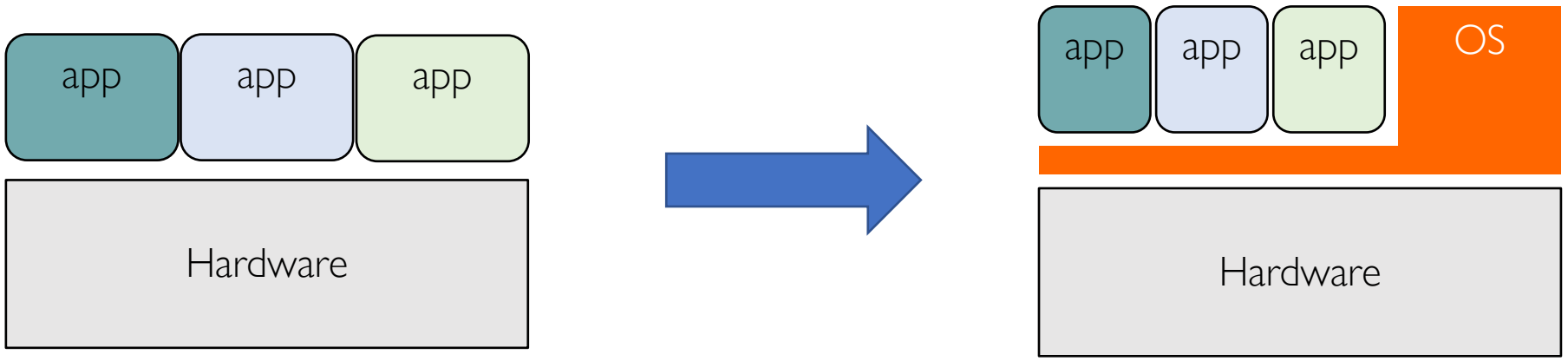


What is an Operating System?

Operating system was once called supervisor or master control program !!!

- A group of software that makes the computer operate **correctly** and **efficiently** in an easy-to-use manner.
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner (**hardware abstraction**)

操作系统隐藏底层硬件的复杂性，给应用提供统一接口



What is an Operating System?

- Which one do you use: MacOS, Windows, or Linux?

Survey:
What OS do you use everyday on your laptop/desktop?

A: Windows
B: Mac OS
C: Linux

What's an Operating System?

- It includes a software program called **kernel** 内核是操作系统的核心软件、直接和硬件打交道
  - manages all the physical devices (e.g., CPU, RAM and hard disk)
  - exposes some functions such as **system calls** for others to configure the kernel or build software (e.g., C library) on top

    其他软件（如 C 标准库、编译器）都依赖内核暴露出来的接口

    提供系统调用（system calls）：
    • 系统调用是应用程序和内核之间的"桥梁"。
    • 比如：一个 C 程序里调用 open("file.txt")，实际上是通过系统调用请求内核帮它访问磁盘。
- It includes other "helper" programs
  - Such as a **shell**, which renders a simple command-line user interface with a full set of commands
  - Such as a **GUI** (graphic user interface), which renders a user-friendly interface with icons representing files and folders
  - Such as a **Browser**, which helps the user to visit websites

    严格来说，浏览器不一定算操作系统的核心，但在现代系统里，它常常被当作"默认工具"，让用户能方便地访问互联网。

What's an Operating System?

- An OS is a **resource manager**
  - Managing CPUs, memory, disks, I/O devices (keyboards, USB drive, sensors, …)
    管理所有硬件资源
  - Arbitrator of conflicting requests for efficient and fair resource use
    当多个程序或用户同时请求同一个资源时，操作系统决定谁先用、用多久
- An OS is a **control program**
  - Controls execution of programs to prevent errors and improper use of the computer
    它不仅分配资源，还要 控制程序的执行，确保系统安全、稳定

What Does an Operating System Do?

- Virtualization
  - Virtualize CPU: Run multiple programs on a single CPU (as if there are many CPUs)
  - Virtualize memory: Give each process (or programs if you will) the illusion of running in its own memory address space
- Concurrency
  - Run multi-threaded programs and make sure they execute correctly
- Persistence
  - Write data (from volatile SRAM/DRAM) into persistent storage
  - Performance, crash-resilience
- ……

在 单个 CPU 上运行多个程序，让用户感觉好像有很多 CPU

通过 进程调度 实现：CPU 在不同程序间快速切换，每个程序都以为自己独占 CPU。

操作系统给每个进程分配一个 独立的内存地址空间，让它觉得自己有“专属内存”。

实际上，所有进程共享同一块物理内存，OS 通过 虚拟内存机制 来隔离和保护

内存（RAM/DRAM）是 易失性存储，断电数据就没了。

操作系统负责把数据写入 持久化存储（磁盘/SSD），保证数据不会丢失。

目标：

- 性能（Performance）：读写速度快。

Evolution of OS

只是一些库函数，主要用来处理底层 I/O 操作（比如读写磁带、打卡片）

- Early OS: just a library to handle low-level I/O
- Atlas computing system: **system calls** that raise the hardware privilege level
  - Special instruction to transit between kernel mode and user mode

引入了 System Calls（系统调用） 的概念。

允许用户程序通过 特权指令 请求内核服务。

- UNIX: support of multi-programming and memory protection
- PC era: better security and useability
  - Disk Operating System (DOS), Mac OS, Windows, Linux
- Smart phones: user-facing applications, more sensors
  - iOS, Android, …

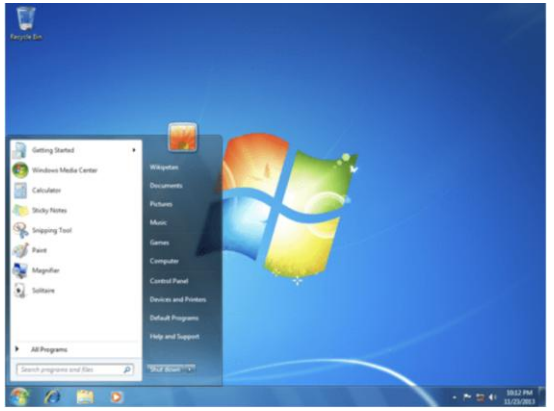
A Brief History of UNIX

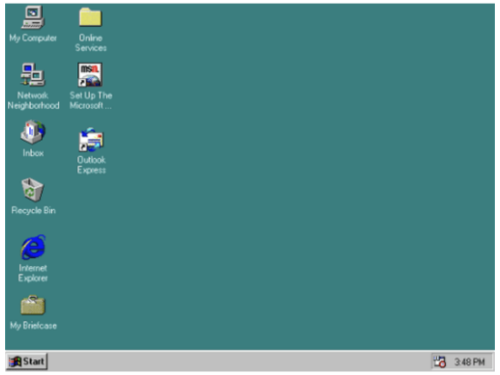
https://web.mit.edu/multics-history/

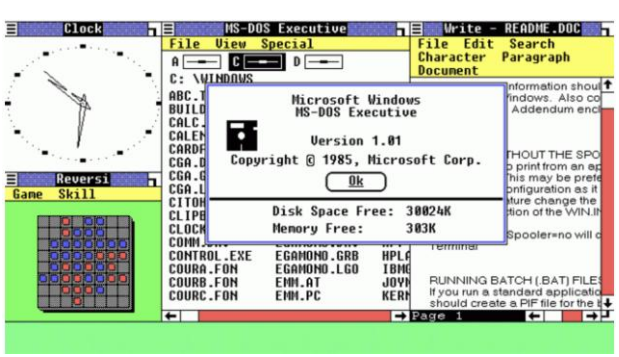
- Influenced by Multics system from MIT
- Originally by Ken Thompson and Dennis Ritchie at Bell Labs
  - Support meta-level programming with shell and pipe
  - Written in easy-to-understand C programming language
- Evolves to Berkeley Systems Distribution (BSD)
  - Advanced virtual memory, file system, and networking subsystems
- Commercial versions of UNIX
  - SunOS from Sun Microsystems, AIX from IBM, HPUX from HP, and IRIX from SGI.
- Mac OS has UNIX at its core
- Ideas and pricinples of UNIX inspire Linus Torvalds
  - The Linux Operating System!! UNIX 的设计思想（“一切皆文件”、小工具组合、进程模型）成为后续操作系统的基石

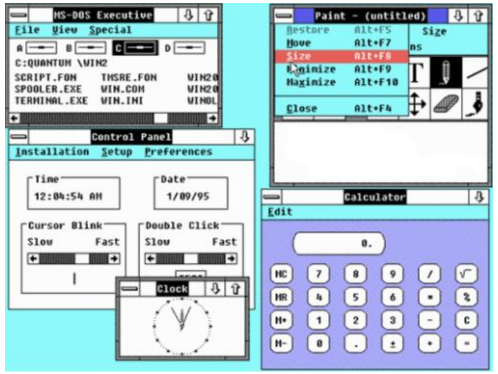
A Brief History of Windows

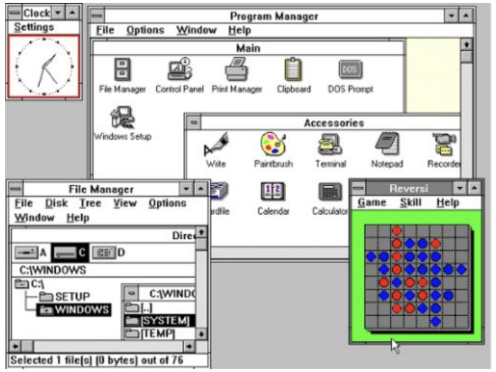
- Windows 1 (1985): Graphic user interface on MS-DOS
- Windows 2 (1987): Support overlaping windows
- Windows 3 (1990): Run MS-DOS programs on Windows
- Windows 3.1 (1992): TrueType fonts support
- Windows 95 (1995): Start menu and button
- Windows 98, ME, 2000, XP, Vista, 7, 8, 8.1, 10

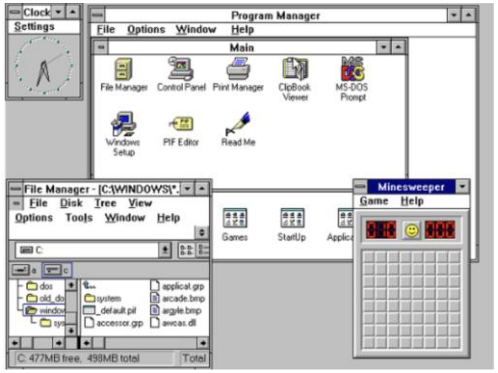












Organization of This Course

Our course will be organzied as a mix of the two

- Organzied by the functionalities of OS (**three easy pieces**)
  - Virtualization (Process, scheduling, memory address space, swapping)
  - Concurrency (Threads, locks, semaphores)
  - Persistence (I/O, storage, file systems)
- Organzied by the resources OS manages (**the dinosaur book**)
  - CPU management (Processes, synchronization, scheduling, deadlocks)
  - Memory management (Physical memory, virtual memory)
  - I/O management (I/O subsystems, storage and file systems)

OS Concept: Processes

- A process is a program in execution
  - Program is a passive entity and process is an active entity.

它只是磁盘上的一个文件（代码+数据），不会自己运行

当程序被加载到内存并执行时，它就成为了一个进程

- Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Process termination requires reclaim of any reusable resources
- Process executes instructions sequentially, one at a time, until completion
  - Single-threaded process has one program counter specifying location of next instruction to execute

每个线程都有自己的 程序计数器，可以在一个进程内部并发执行多个任务

  - Multi-threaded process has one program counter per thread
- Typically, system has many processes, some user, some operating system running concurrently on one or more CPUs
  - Concurrency by multiplexing the CPUs among the processes / threads

并发（Concurrency）的实现方法是：操作系统把 CPU 时间片分配给不同的进程/线程（也就是时间复用）

OS Concept: Process Management

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

OS Concept: Memory

因为DRAM速度快

- DRAM (Dynamic Random Access Memory) is the main memory used for all desktop, laptops, servers, and mobile devices
- CPU only directly interacts with the main memory during execution
  - All data in memory before and after processing
  - All instructions in memory in order to execute
- OS manages the main memory for kernel and processes
  - OS dictates which process can access which memory region

CPU 在执行时只能 直接访问主存：

- 所有要运行的数据（输入、计算结果）都必须先放到内存里。
- 所有要执行的指令（程序代码）也必须在内存里。

操作系统负责给不同的进程分配内存，决定哪个进程能访问内存的哪一部分。

这样避免进程之间互相干扰，保证安全和稳定

OS Concept: Memory Management

- Memory management determines what is in memory when
  - Optimizing CPU utilization and computer response to users
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating memory space as needed

决定数据和进程的调度

- 决定哪些进程（或进程的一部分，比如某个函数、某些数据）需要放到内存中。
- 决定什么时候把数据换出到磁盘（swap），什么时候再换入。

OS Concept: Storage Management

操作系统给用户提供的是一个 逻辑上的统一视图，而不是直接面对硬件的复杂性

- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - file
  - Each medium is controlled by device (i.e., disk drive, tape drive)
    - Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)

把底层的物理存储抽象成“文件”

- File-System management
  - Files usually organized into directories
  - Access control on most systems to determine who can access what
  - OS activities include
    - Creating and deleting files and directories
    - Primitives to manipulate files and dirs
    - Mapping files onto secondary storage
    - Backup files onto stable (non-volatile) storage media

创建和删除文件/目录

• 用户 touch file.txt 或者 mkdir docs 背后，OS 负责在磁盘上分配空间并更新元数据。

提供基本操作（primitives）

• 例如打开文件、读写数据、关闭文件。

文件映射（Mapping）

• OS 需要把逻辑文件映射到物理存储上的某些块（blocks/sectors）。

• 比如文件大小 10MB，可能被拆分存在硬盘不同位置，但对用户来说仍然是一个完整文件。

备份（Backup）

• OS 还会提供备份机制，把文件保存到更稳定、持久的介质上（非易失性存储）。

• 例如快照（snapshot）、系统还原点、磁带存档。

OS Concept: I/O Subsystem

- One purpose of OS is to hide peculiarities of hardware devices from the user

隐藏硬件的复杂性，给用户和应用一个统一的接口

- I/O subsystem responsible for
  - Memory management of I/O including
    - buffering (storing data temporarily while it is being transferred)
    - caching (storing parts of data in faster storage for performance)
  - General device-driver interface
  - Drivers for specific hardware devices

缓冲（buffering）

- 在数据传输过程中，先放到一个临时区域。
- 举例：你复制一个 1GB 文件，数据不会直接“一下子”传到硬盘，而是先进入缓冲区再慢慢写入。

缓存（caching）

- 把常用的数据放到更快的存储里，加快访问速度。
- 举例：CPU 访问硬盘文件时，常见的数据会被缓存到内存里。

OS Concept: Protection and Security

保护机制是 内部规则，确保即使在正常使用场景下，资源访问也被严格管控。

- **Protection** – any mechanism for controlling access of processes or users to resources defined by the OS
- **Security** – defense of the system against internal and external attacks
  - Huge range, including denial-of-service, worms, viruses, identity theft, theft of service

安全机制是 外部防御、主要应对恶意攻击。

- OS determines which users can do what
  - User identities (user IDs, security IDs) include name and associated number, one per user
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (group ID) allows set of users to be defined and controls managed, then also associated with each process, file
  - Privilege escalation allows user to change to effective ID with more rights

Thank you!