Introduction to Operating Systems

CSD2180 Operating Systems
BSc in Computer Science (IMGD / RTIS)
Singapore Institute of Technology / DigiPen Institute of Technology
September 2021

Adapted from Operating Systems Concepts - 10th Edition; Silberschatz, Galvin, and Gagne. ©2018

What Operating Systems Do | Computer-System Organization | Computer-System Architecture | Operating-System Operations | Resource Management | Security and Protection | Virtualization | Distributed Systems | Kernel Data Structures | Computing Environments | Free and Open-Source Operating Systems

What is an Operating System?

A program that acts as an intermediary between a user of a computer and the computer hardware

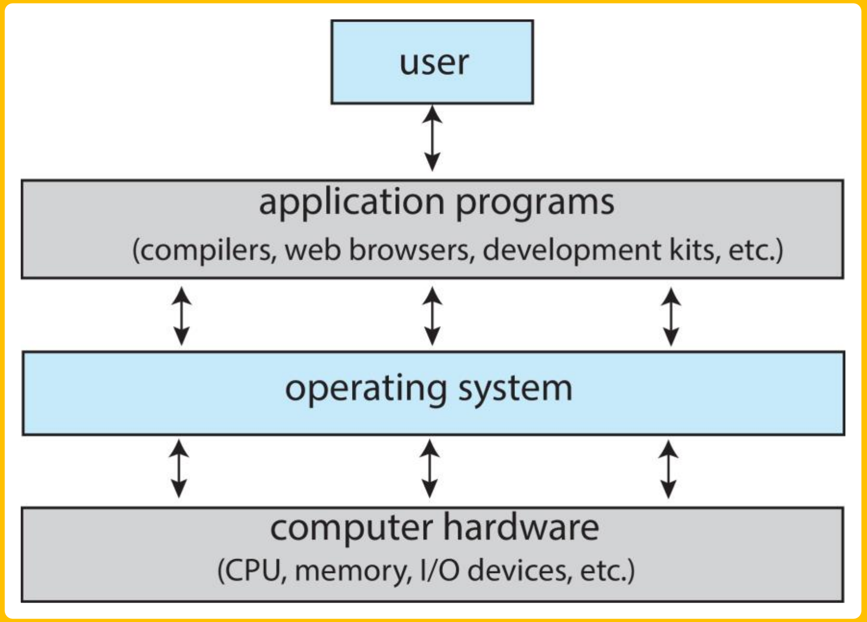
Operating system goals

- Execute user programs and make solving user problems easier
- Make the computer system convenient to use
- Use the computer hardware in an efficient manner

"Wei macha, you want to makan or tapau?"

00110010001100100011000100110011

Computer System Structure



Hardware: Provides basic computing resources

Operating system: Controls / 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

Users: People (you!), machines, other computers

What do operating systems do?

What Do Operating Systems Do?

Users want convenience, **ease of use** and **good performance**

Shared computer such as mainframe or minicomputer must **keep all users happy**

Users of dedicate systems such as workstations have dedicated resources but frequently use **shared resources** from servers

Mobile devices like smartphones and tables are resource poor, optimized for **usability** and **battery life**

Some embedded computers have little or no user interface

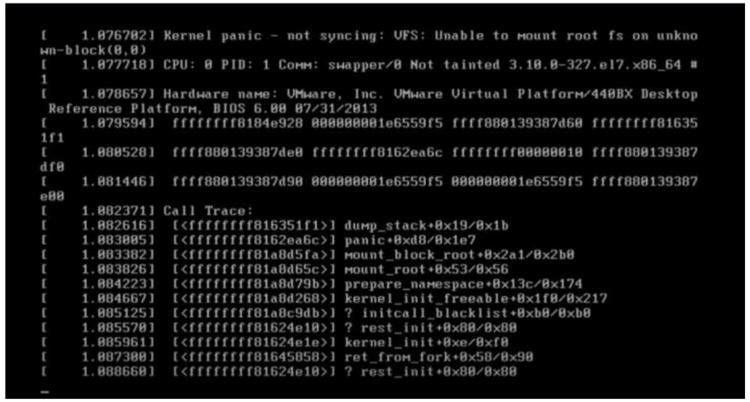
Depends on the point of view

*Operating system is a **resource allocator** and **control program**, making efficient use of hardware and managing execution of user programs*

What is an operating system, really?

The Definition of Operating System

"Everything a vendor ships when you order an operating system"



Kernel

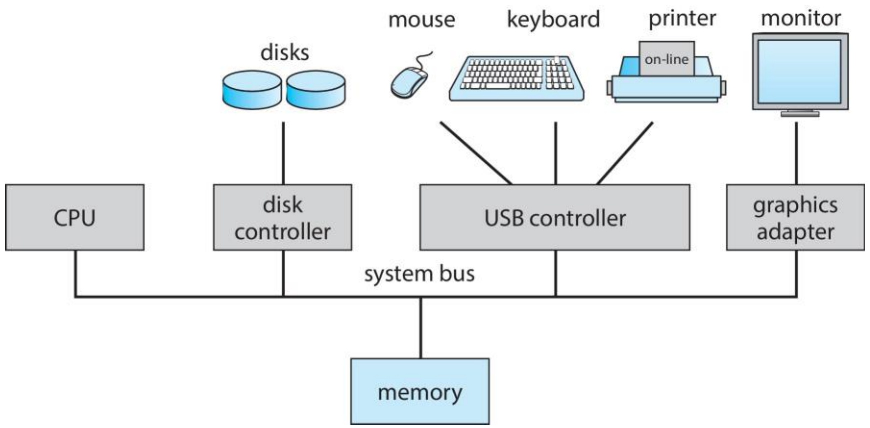
- Part of the operating system
- The one program always running on the computer

Everything else is either

- A **system program** (ships with the operating system, but not part of the kernel), or
- An **application program**, all programs not associated with the operating system

Computer-System Organization

Computer-System Operation



One or more CPUs / device controllers connect through a **common bus** providing access to shared memory

Concurrent execution of CPUs and devices competing for memory cycles

I/O devices and the CPU can execute concurrently

Each device controller

- Oversees a particular device type, e.g., disks, etc.
- Has a local buffer, i.e., temporary data holding area
- Has an operating system device driver to manage it

CPU moves data from / to main memory to / from local buffers

I/O is from the device to local buffer of controller

Device controller informs CPU that it has finished its operation by causing an **interrupt**

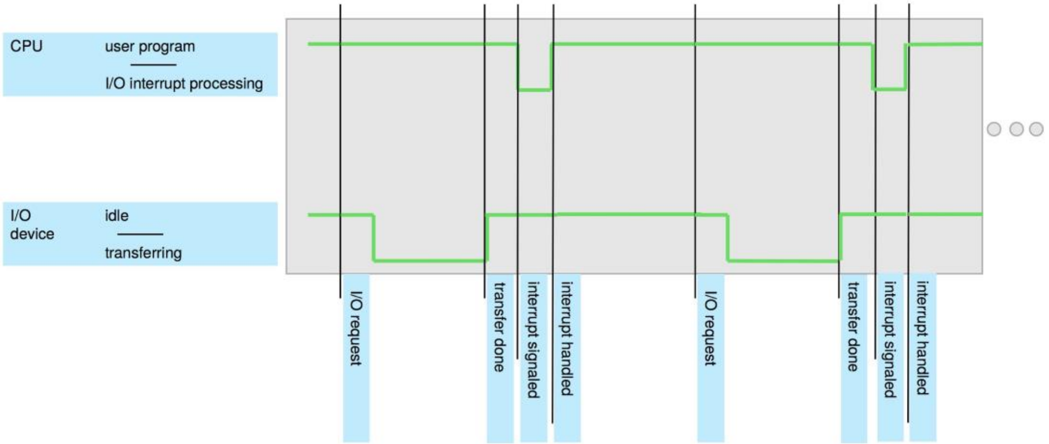
Interrupts

An operating system is **interrupt driven**

Generally, interrupt transfers control to the **interrupt service routine**, through the **interrupt vector**, which contains the addresses of all the service routines

Interrupt architecture must save the address of the interrupted instruction

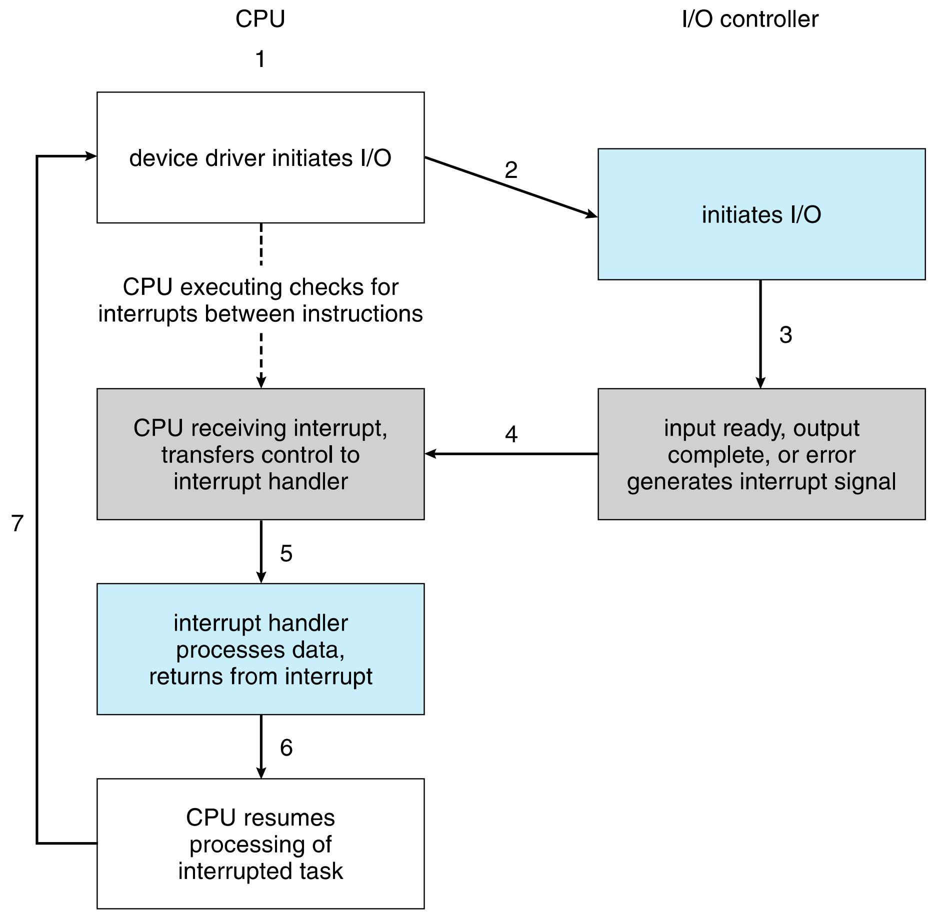
A **trap** or **exception** is a software-generated interrupt caused either by an error or a user request



Interrupt-Driven I/O Cycle

Interrupt handling in a nutshell

- The operating system preserves the state of the CPU by storing the registers and the program counter
- Determines which type of interrupt has occurred
- Separate segments of code determine what action should be taken for each type of interrupt



Storage Structure

Main memory

Large storage media that CPU can access directly

Random access, typically volatile

Typically in the form of Dynamic Random-Access Memory (DRAM)

Secondary storage

Extension of main memory that provides large nonvolatile storage capacity

Hard Disk Drives

Mechanical data storage device

Stores data on **rotating platter(s)** within the disk body

Read / write performed by a **head** on a platter

The head is connected to an **arm**, which is moved by an **actuator**

Movement of the arm is not done using motor, instead using a **voice coil**

Solid State Drives

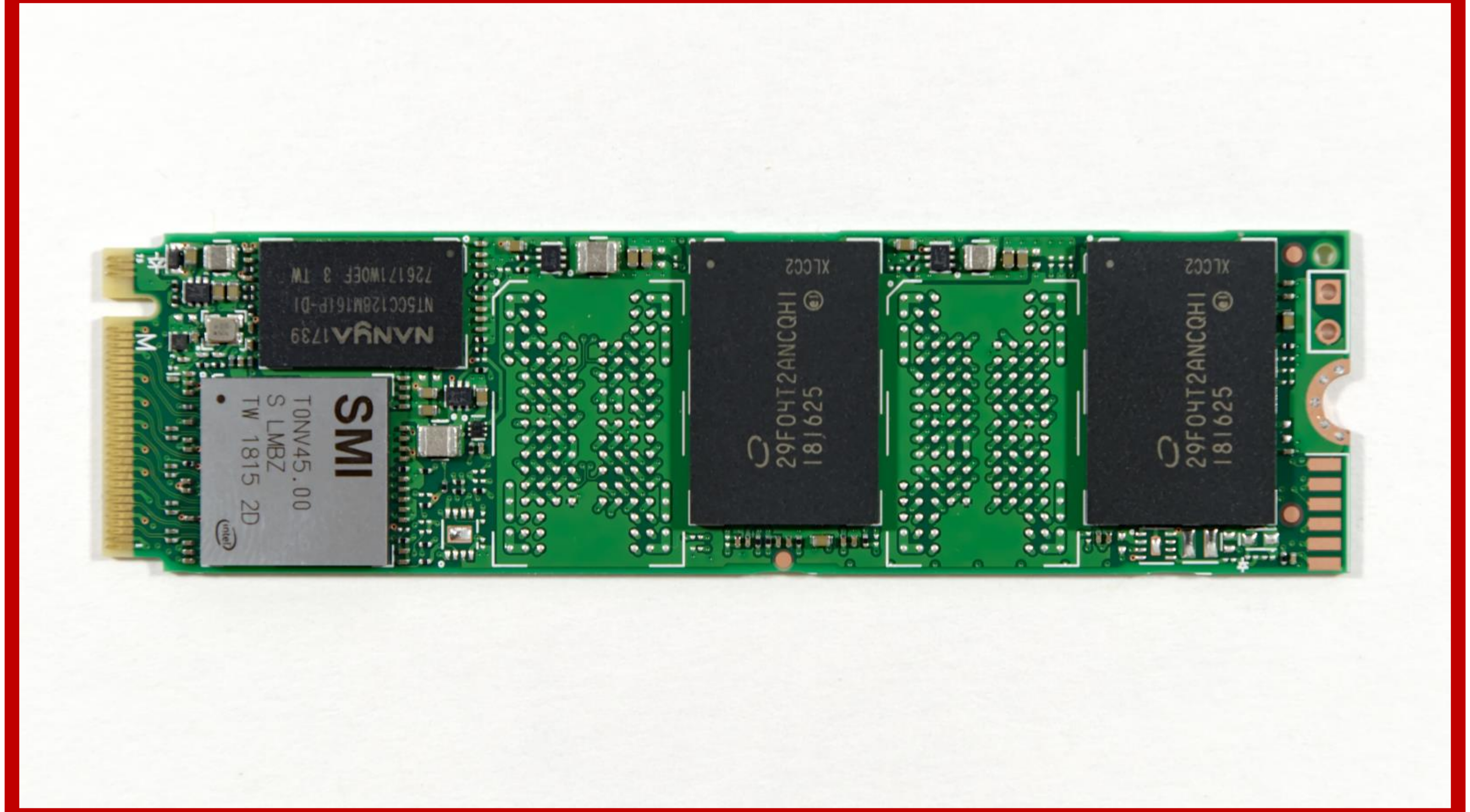
Entire drive is **electronic**, with no mechanical components

Much **faster** than a hard disk

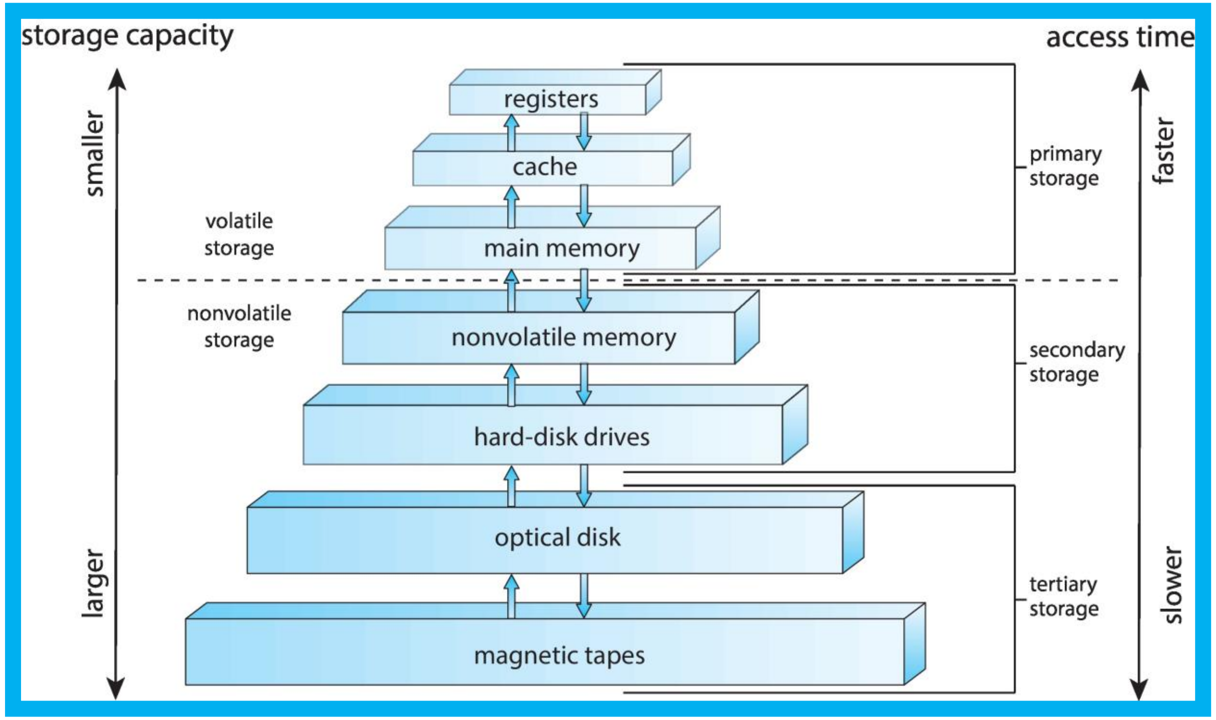
Typically utilize **NAND flash memory** to store contents

Flash memory is managed by the **SSD controller**, which also manages the drive's interface

Some SSDs have **DRAM** to improve performance



Storage Device Hierarchy



Level	1	2	3	4	5
Name	registers	cache	main memory	solid-state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25-0.5	0.5-25	80-250	25,000-50,000	5,000,000
Bandwidth (MB/sec)	20,000-100,000	5,000-10,000	1,000-5,000	500	20-150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

Some values may not be that accurate today...

Protip: Be careful about how megabytes, kilobytes, etc. are represented

Why Are We Talking About Storage?

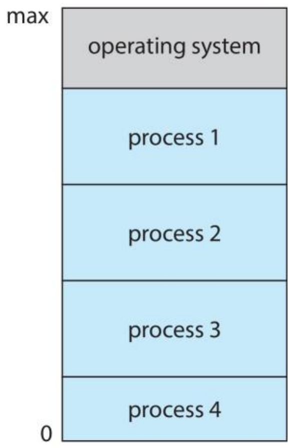
Memory Management

To execute a program all (or part) of the instructions must be in memory

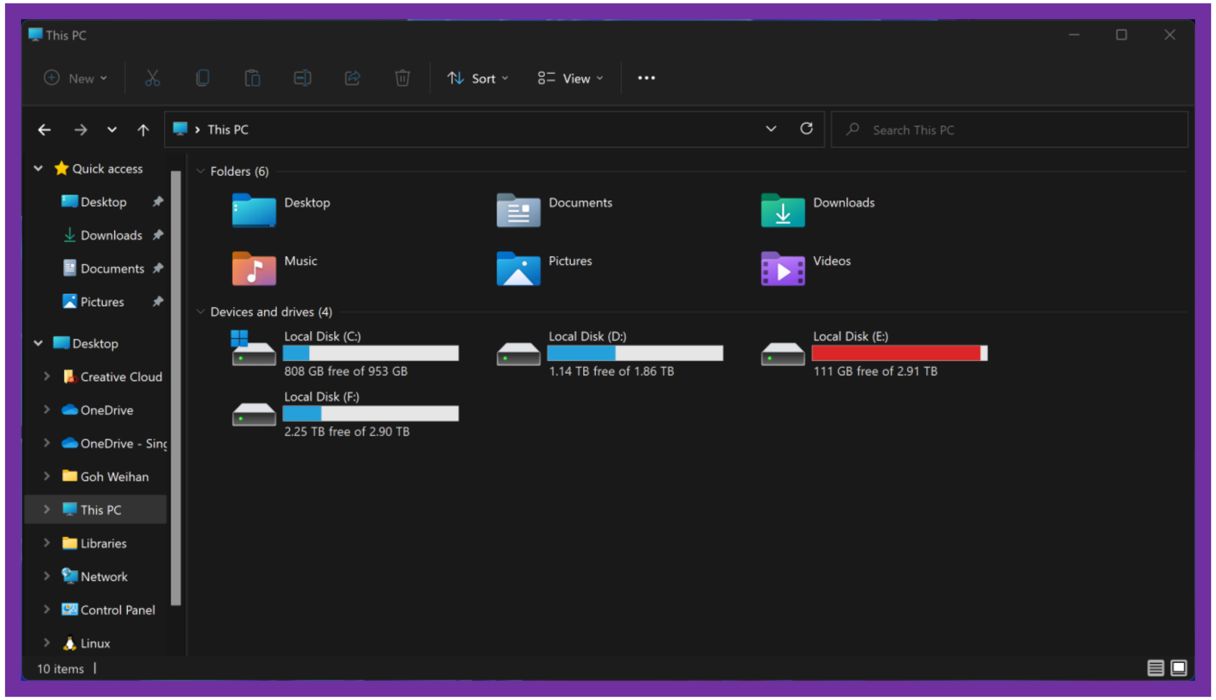
Memory management determines what is in memory and when

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



Why Are We Talking About Storage?



Filesystem Management

Operating system provides uniform, logical view of information storage

Filesystem management

- Files usually organized into directories
- Access control on most systems to determine who can access what
- Operating system activities include
  - 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

Why Are We Talking About Storage?

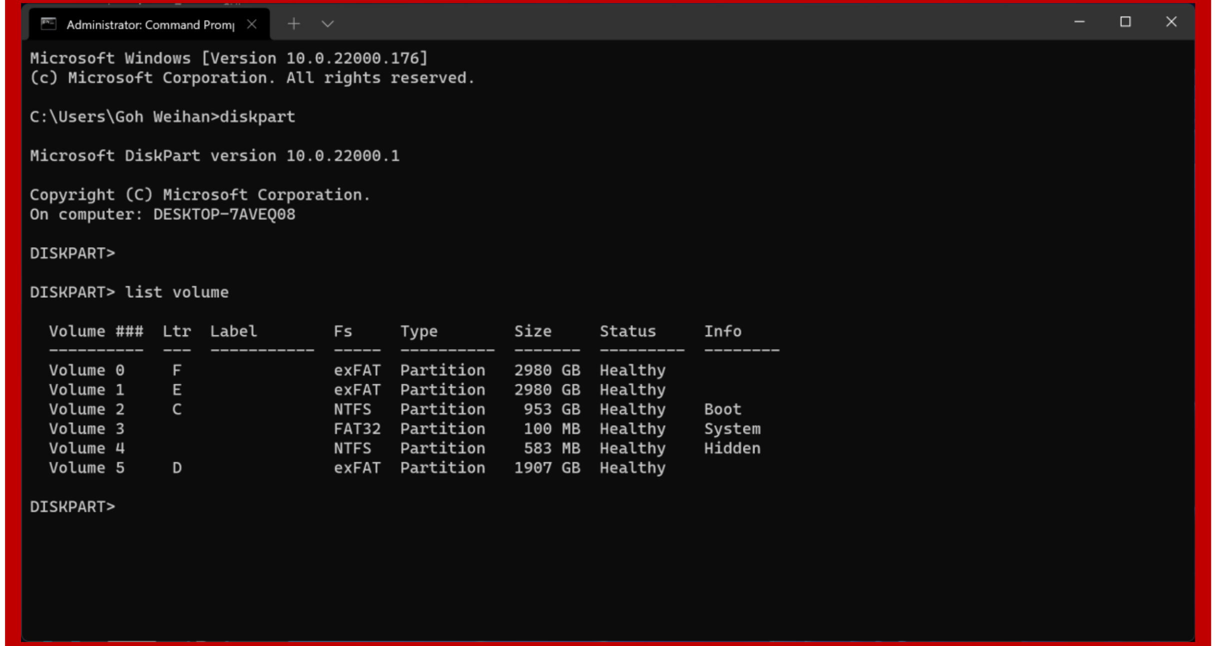
Mass-Storage Management

Usually, 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

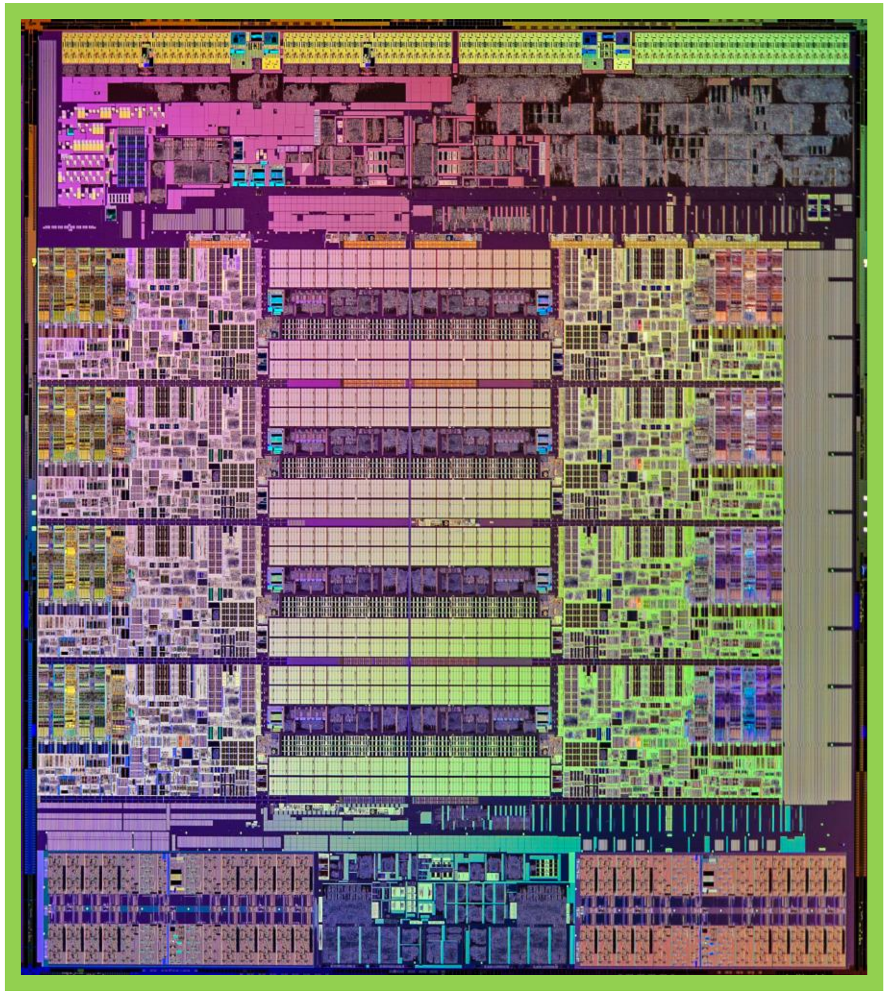
Proper management is of central importance

Operating system activities

- Mounting and unmounting (i.e., safe removal)
- Free-space management
- Storage allocation
- Disk scheduling
- Partitioning
- Protection



Why Are We Talking About Storage?



Caching

Information in use copied from slower to faster storage temporarily

Faster storage (cache) checked first to determine if information is there

- If it is, information used directly from cache (fast)
- If not, data copied to cache and used there

Cache smaller than storage being cached

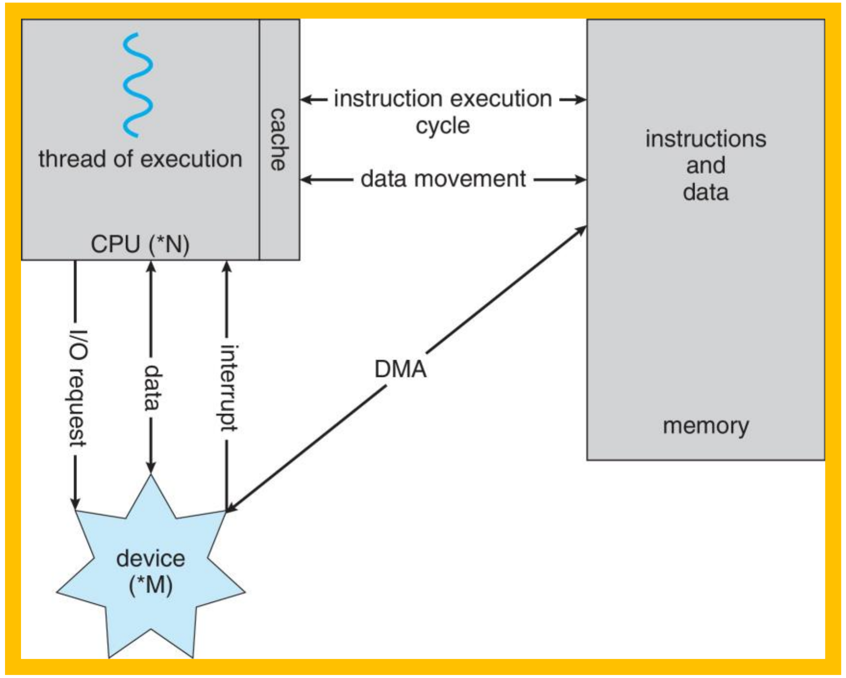
- Cache management important design problem
- Cache size and replacement policy

Direct Memory Access

Used for high-speed I/O devices able to transmit information at close to memory speeds

Device controller transfers blocks of data from buffer storage directly to main memory without CPU intervention

Only one interrupt is generated per block, rather than the one interrupt per byte



Two Methods for Handling I/O

Synchronous / Blocking I/O

After I/O starts, control returns to user program only upon I/O completion

(CPU stays idle and waits until it receives an interrupt after I/O completion)

Asynchronous / Non-Blocking I/O

After I/O starts, control returns to user program without waiting for I/O completion

(More efficient, CPU can do other things while I/O is happening)

Computer-System Architecture

Computer System Architecture

Most systems use a single **general-purpose processor**

Some systems have **special-purpose processors** as well, e.g., for security, AI, etc.

Multiprocessor systems are ubiquitous today

- Also known as parallel systems, tightly-coupled systems
- Not uncommon to find systems with **multiple CPUs or CPU cores**

Advantages of multiprocessor systems

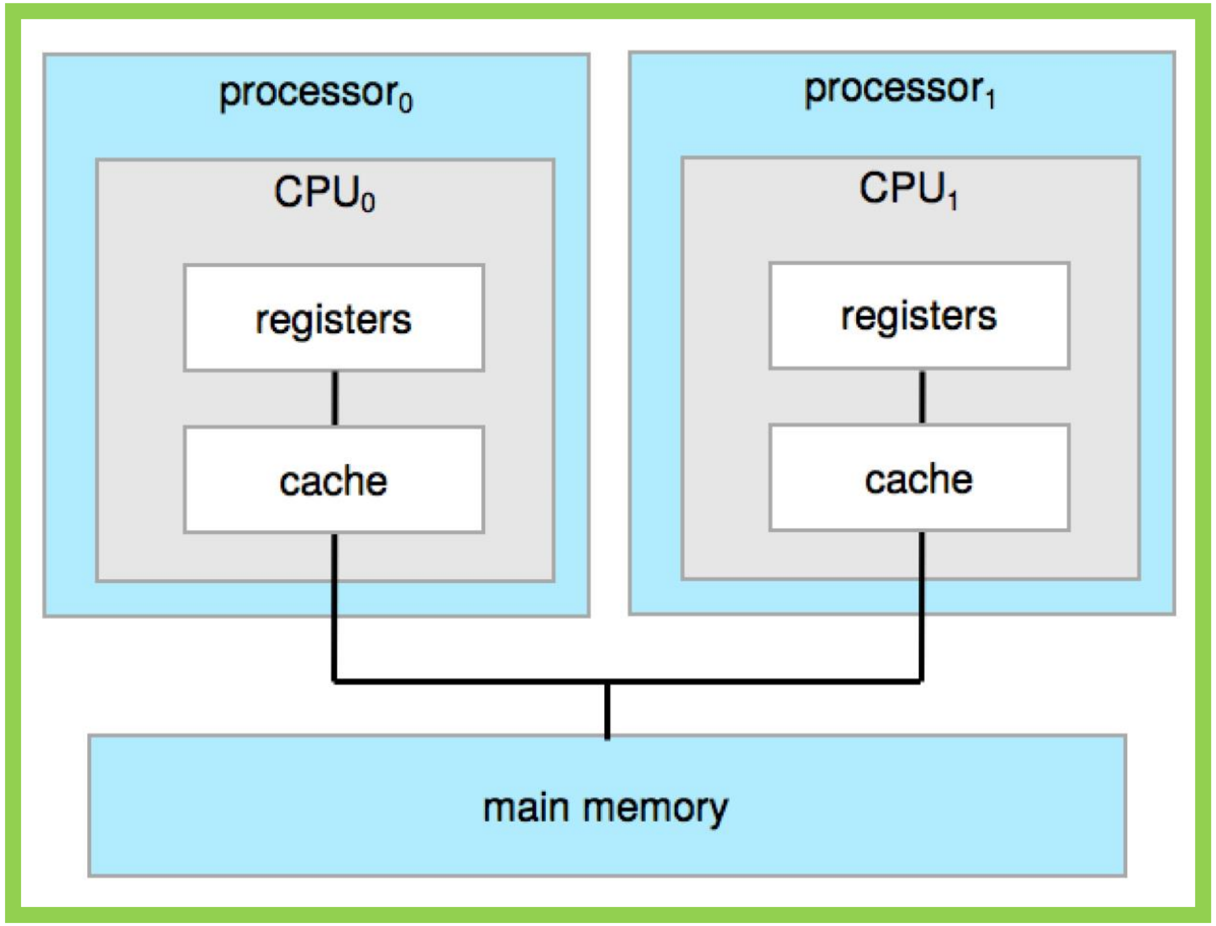
- **Increased throughput:** There is a speed-up, but some overhead is incurred to get everything working correctly; this overhead plus contention for shared resources lowers the expected gain
- **Economy of scale:** Sharing of peripherals, mass storage, power supplies, etc.
- **Increased reliability:** Graceful degradation or fault tolerance; if functions are properly distributed across the processors, then the failure of any one subsystem will not halt a system

Computer System Architecture

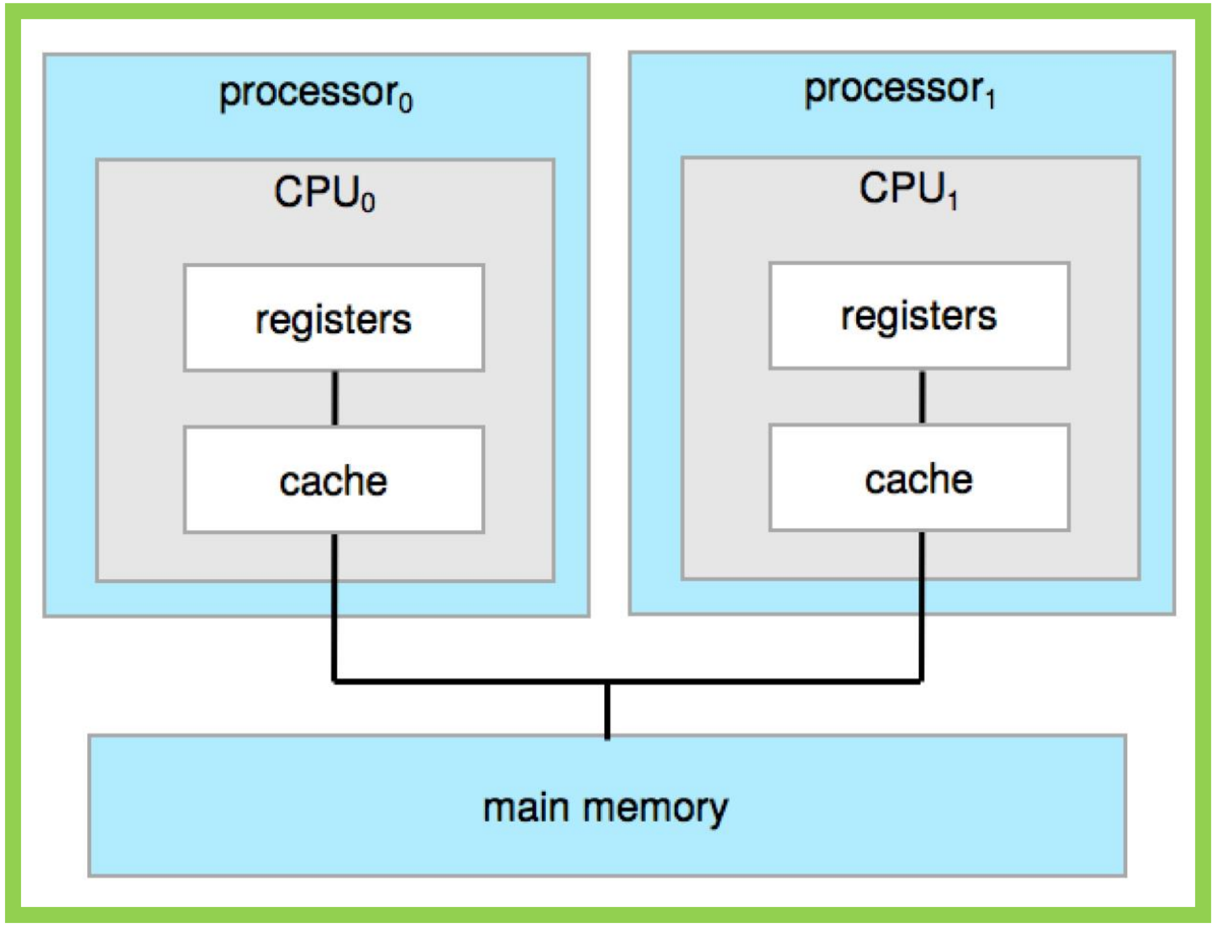
Two types of multiprocessor system

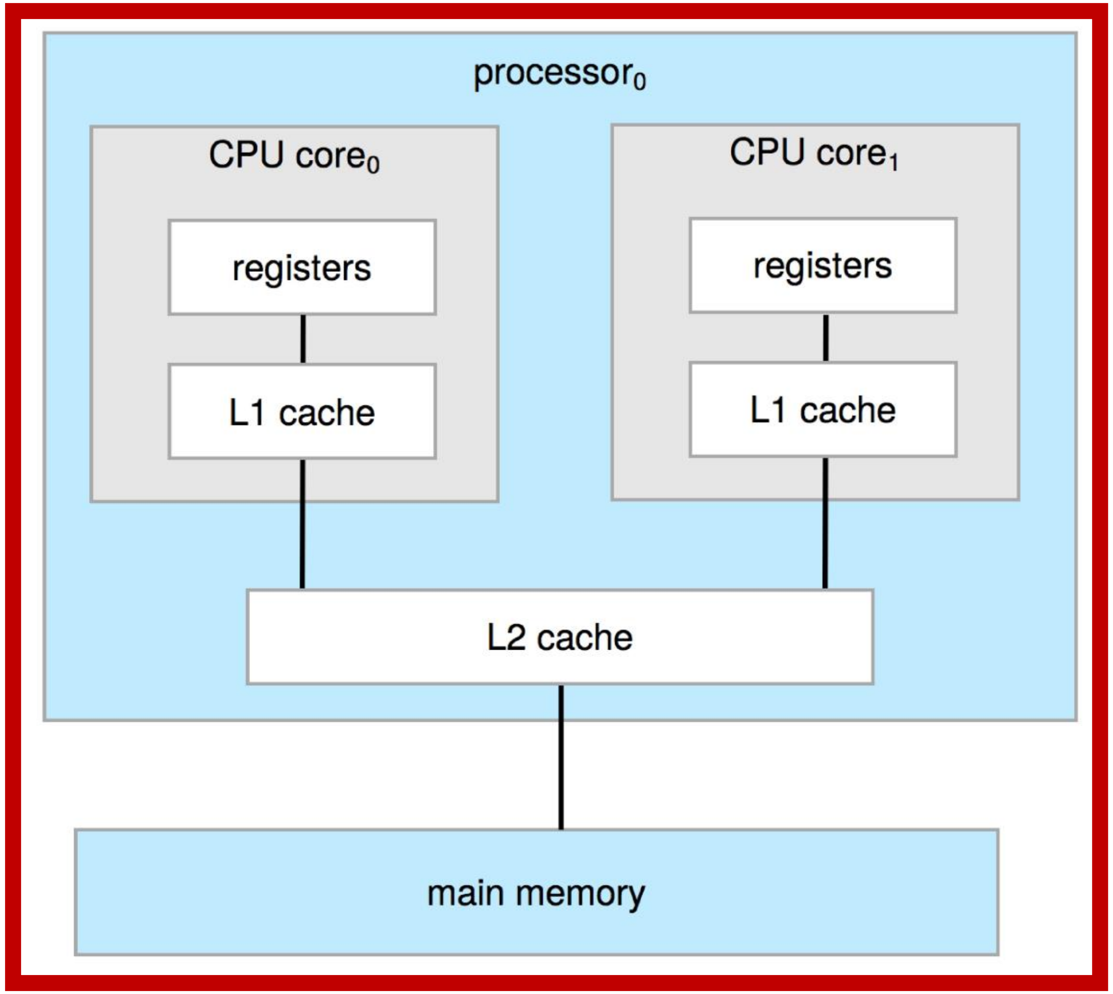
Asymmetric multiprocessing: Less common; each processor is assigned a specific task; a master processor controls the system and schedules and allocates work to the other processors

Symmetric multiprocessing: More common; each processor can perform *any* tasks within the operating system; many processes can run simultaneously but with some inefficiencies



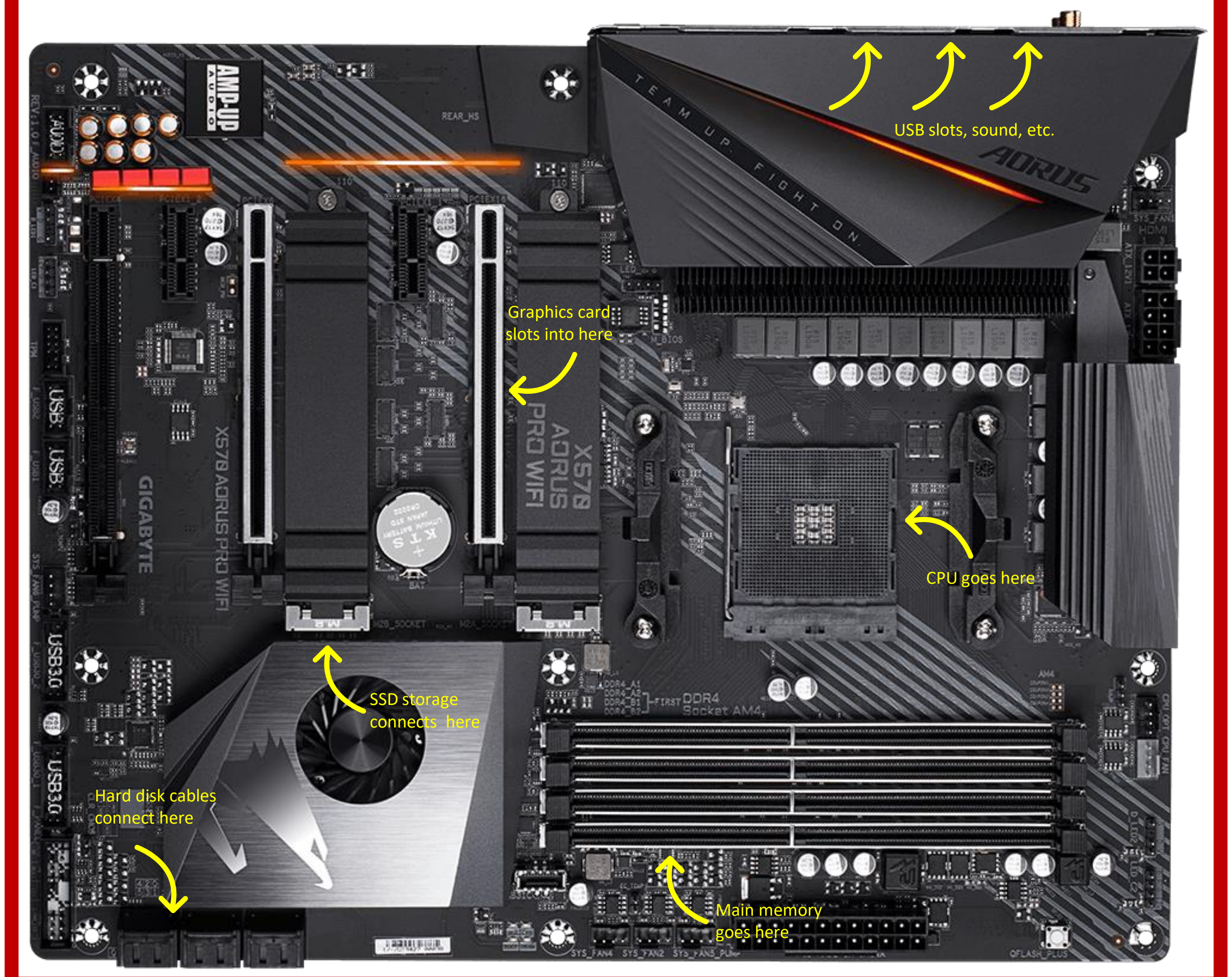
Multiple Chip and Multicore Systems



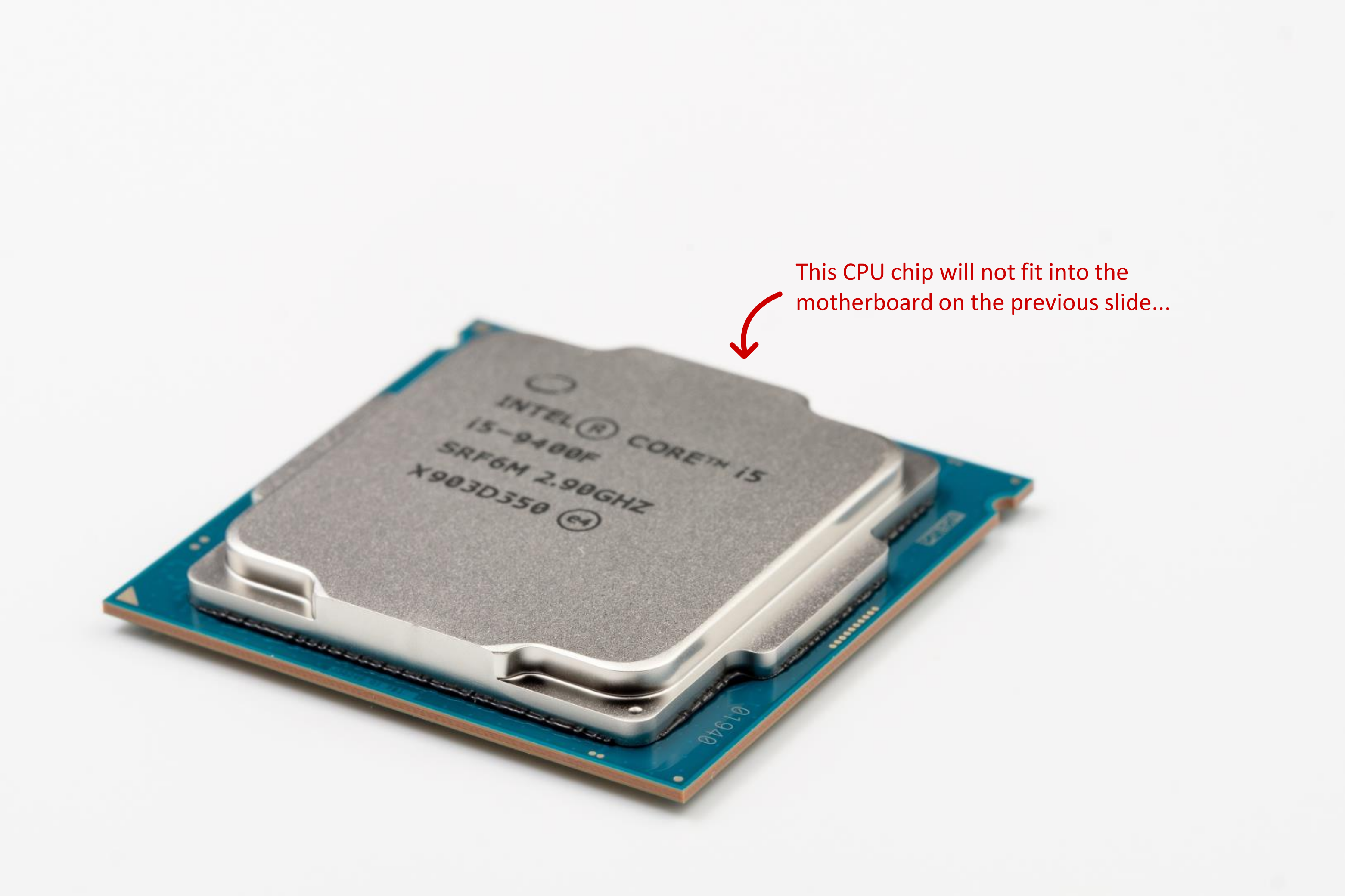


Multicore can be more efficient than multiple chips with single cores as they consume less power overall; also, on-chip communication is faster

A Modern Computer Motherboard



A Modern CPU Chip



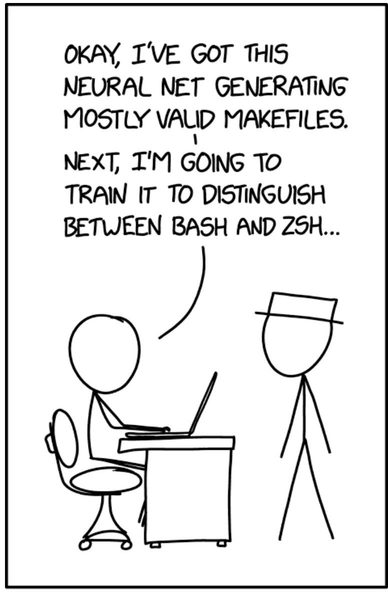
Questions?

Thank You!

Weihan.Goh {at} Singaporetech.edu.sg

https://www.singaporetech.edu.sg/directory/faculty/weihan-goh

https://sg.linkedin.com/in/weihan-goh



PEOPLE OFTEN USE ANCIENT TOOLS AND UIs TO DEVELOP MODERN CUTTING-EDGE TECHNOLOGY, BUT I DO IT THE OTHER WAY AROUND.