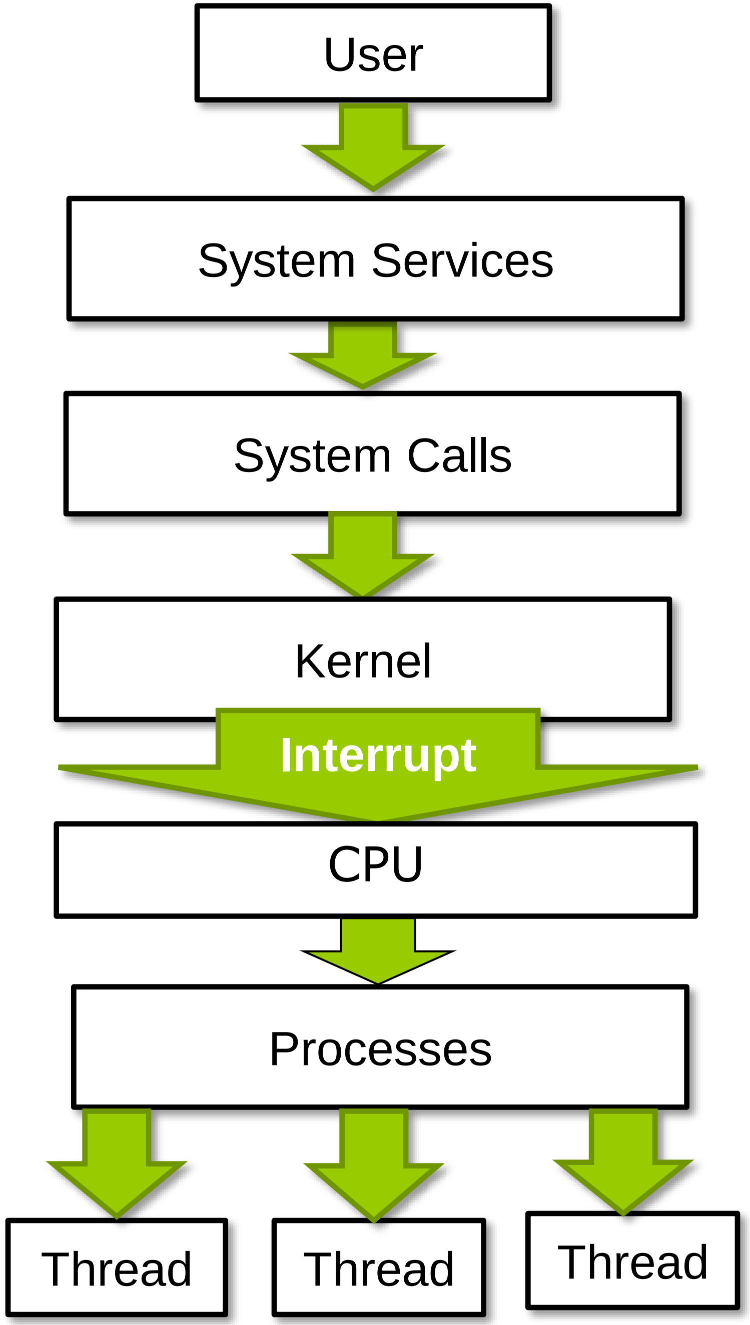
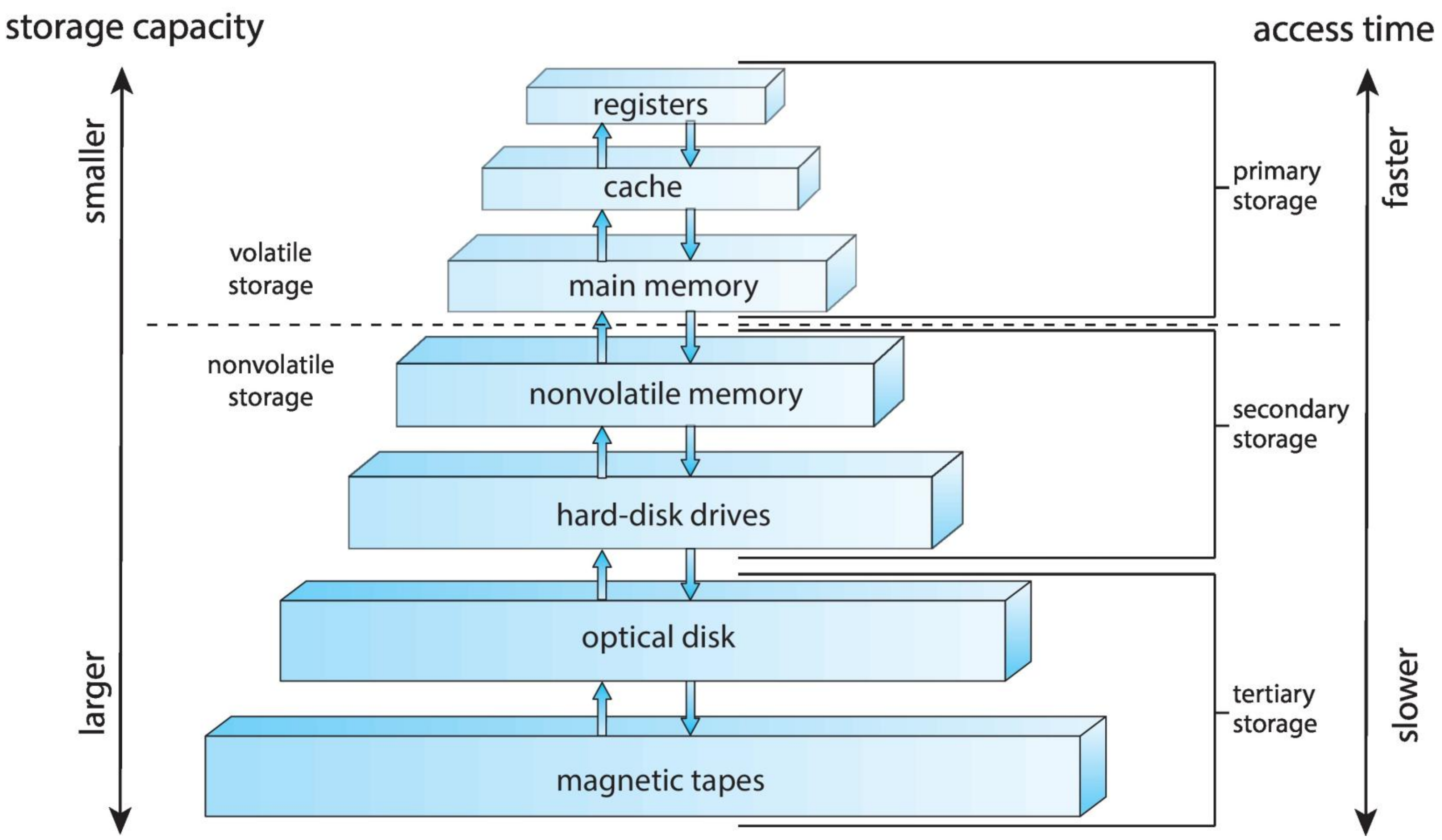
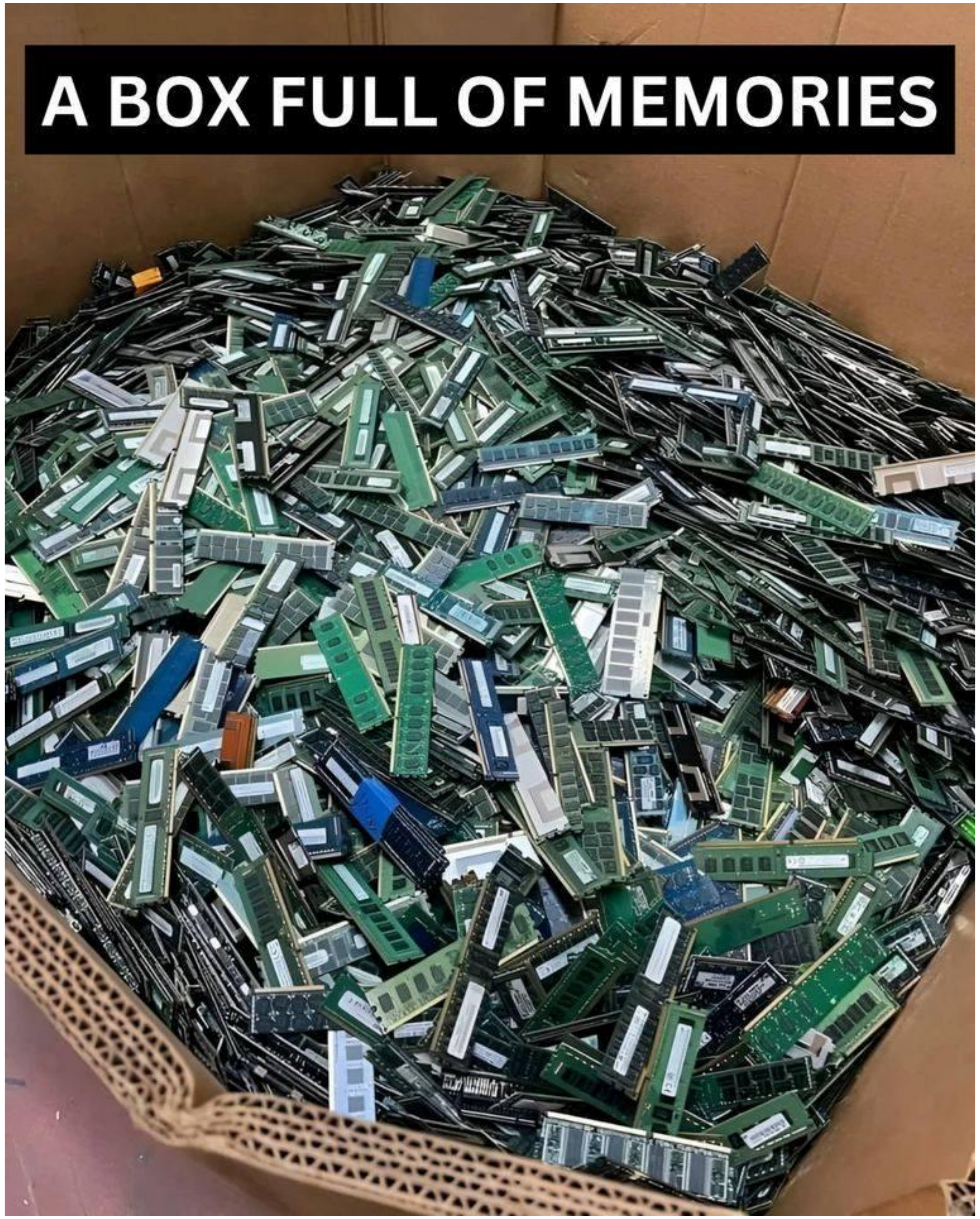
Part 3

The Big Pic of OS

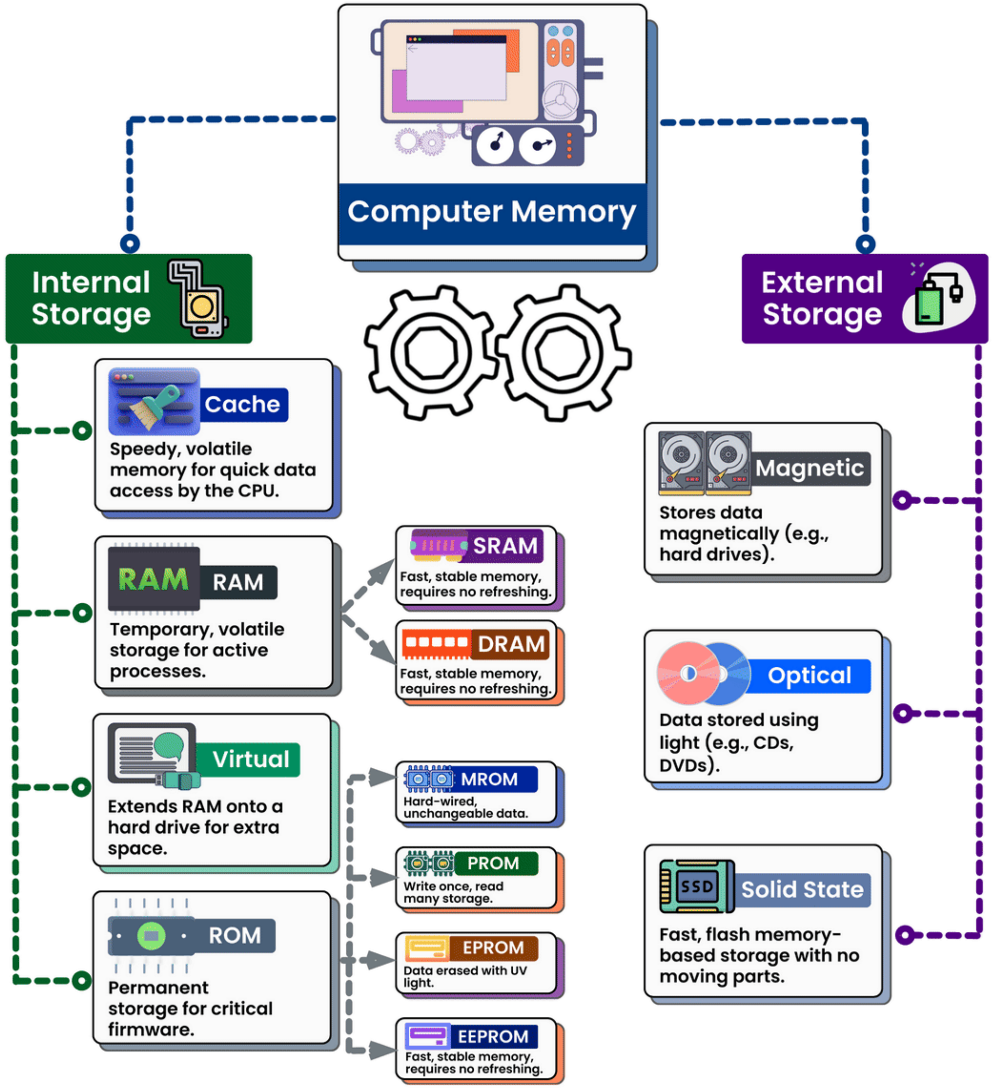


Storage-Device Hierarchy

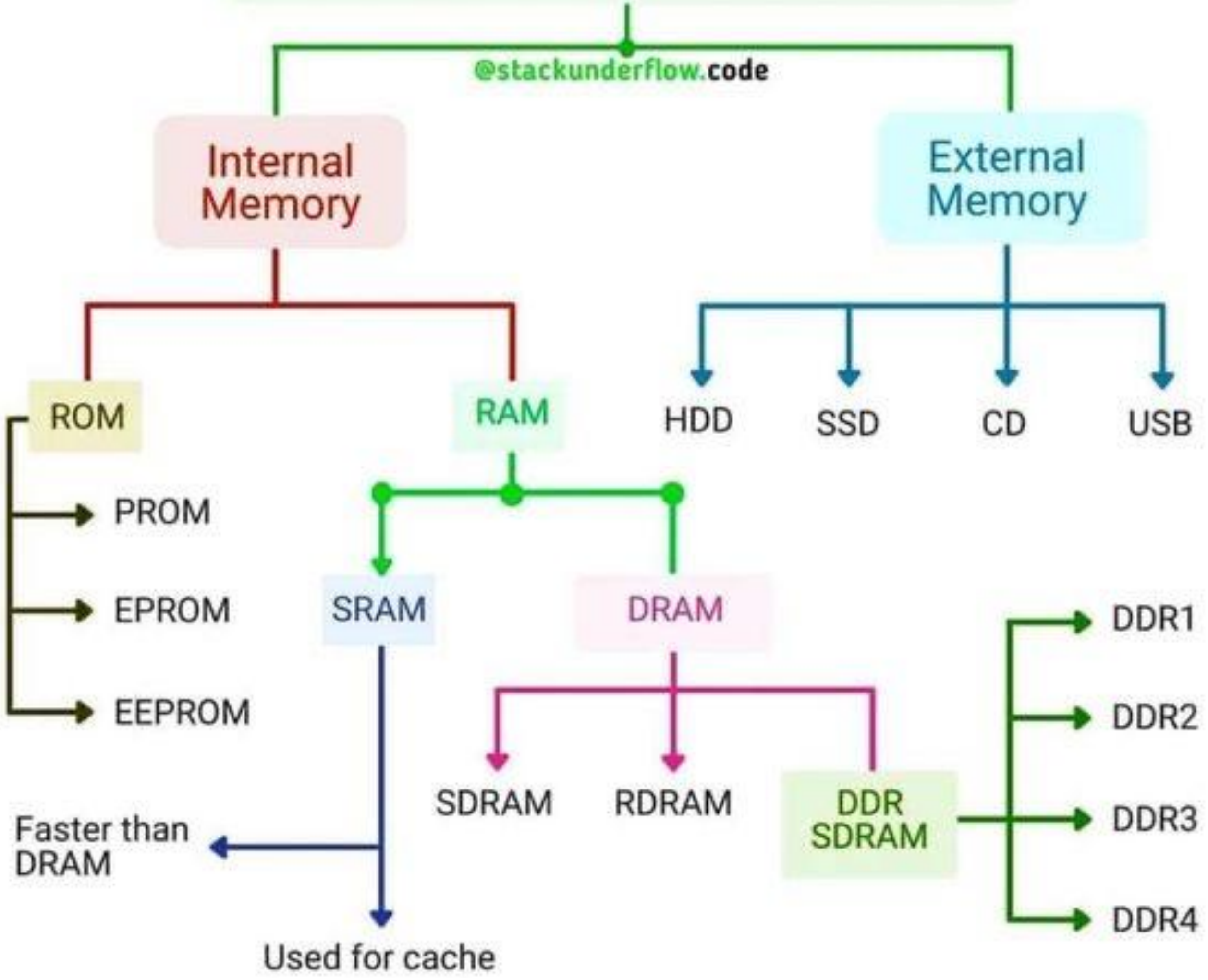



A BOX FULL OF MEMORIES

Types of Memory
In Computer



TYPES OF Computer Memory



Storage Hierarchy

- Storage systems organized in hierarchy
  - Speed
  - Cost
  - Volatility
- **Caching** – copying information into faster storage system; main memory can be viewed as a cache for secondary storage
- **Device Driver** for each device controller to manage I/O
  - Provides uniform interface between controller and kernel

Direct Memory Access Structure

- 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

Computer-System Architecture

- Most systems use a single general-purpose processor
  - Most systems have special-purpose processors as well
- **Multiprocessors** systems growing in use and importance
  - Also known as **parallel systems**, **tightly-coupled systems**
  - Advantages include:
    1. **Increased throughput**
    2. **Economy of scale**
    3. **Increased reliability** – graceful degradation or fault tolerance
  - Two types:
    1. **Asymmetric Multiprocessing** – each processor is assigned a specie task.
    2. **Symmetric Multiprocessing** – each processor performs all tasks

RAM vs. ROM

- RAM (Random Access Memory) and ROM (Read-Only Memory) are two types of computer memory that serve different purposes in a computer system. Here are the key differences between them:

Volatility:

1. RAM is volatile memory, which means it loses its data when the computer is powered off or restarted. It is used for temporary data storage while the computer is running.
2. ROM is non-volatile memory, meaning it retains its data even when the computer is turned off. It stores essential data and instructions that are required for booting up the computer and initializing hardware components.

Function:

- RAM is used for temporary storage of data that the CPU (Central Processing Unit) actively uses during the execution of programs and tasks. It provides fast read and write access and is crucial for the computer's performance.
- ROM contains firmware or permanent software instructions that are integral to the computer's operation, such as the BIOS (Basic Input/Output System) or UEFI (Unified Extensible Firmware Interface). It is responsible for the initial boot-up of the computer and the loading of the operating system.

Read/Write Access:

- RAM is read-write memory, allowing data to be both read from and written to it. This makes it suitable for tasks like running applications, storing temporary files, and managing the computer's active processes.
- ROM is read-only memory, and its contents are typically programmed during the manufacturing process and cannot be easily altered or updated by the user. It is meant for storing permanent data and instructions.

Size and Capacity:

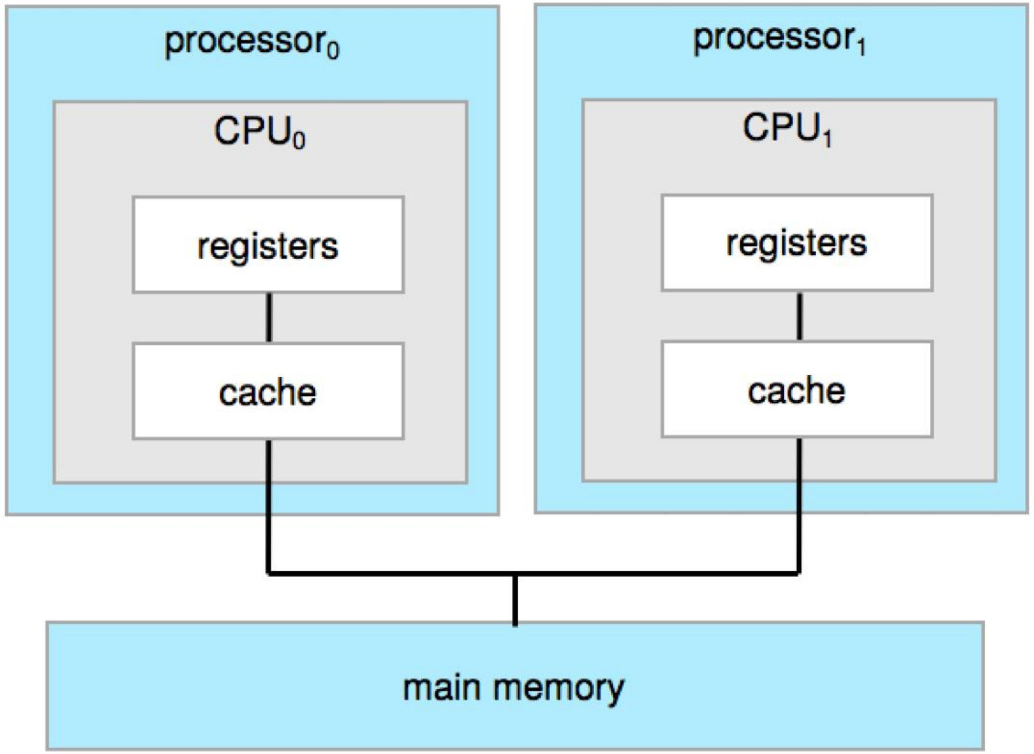
- RAM is usually larger in terms of capacity than ROM in modern computers. RAM capacity can range from a few gigabytes to several terabytes.
- ROM is typically smaller in capacity compared to RAM, as it stores essential firmware and boot-up instructions. It is usually measured in megabytes or gigabytes.

Access Speed:

- RAM offers much faster access speeds compared to ROM. This high-speed access allows the CPU to quickly read and write data during program execution.
- ROM provides slower access speeds compared to RAM because it contains firmware that is not meant to be frequently accessed during normal operation.

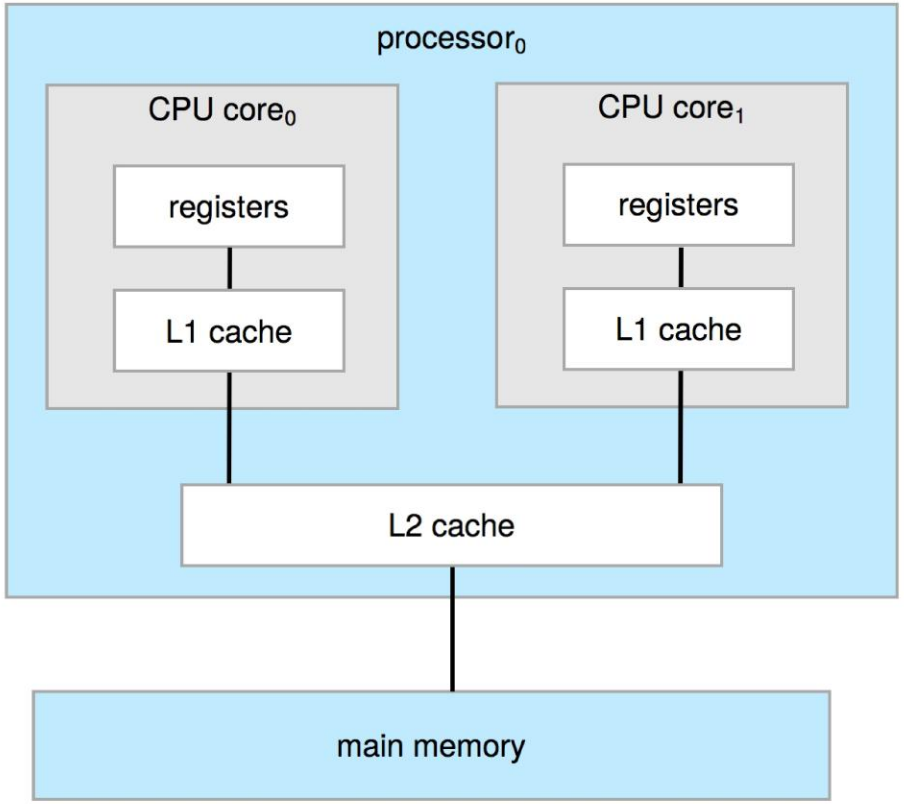
- RAM and ROM serve distinct roles in a computer system.
- RAM provides temporary storage for actively running programs and data, while ROM contains permanent instructions and firmware essential for booting up the computer and initializing hardware components.
- These differences in functionality, volatility, and access speeds make them essential components in modern computing devices.

Symmetric Multiprocessing Architecture

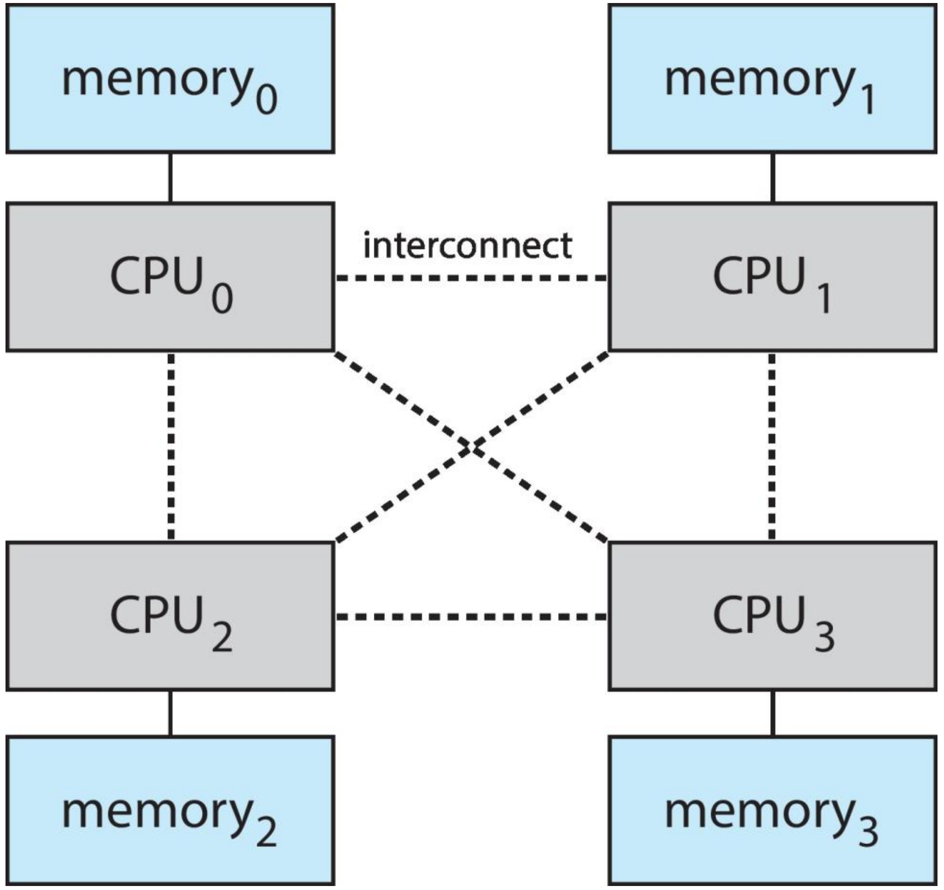


A Dual-Core Design

- Multi-chip and **multicore**
- Systems containing all chips
  - Chassis containing multiple separate systems



Non-Uniform Memory Access System



Clustered Systems

- Like multiprocessor systems, but multiple systems working together
  - Usually sharing storage via a **storage-area network (SAN)**
  - Provides a **high-availability** service which survives failures
    - **Asymmetric clustering** has one machine in hot-standby mode
    - **Symmetric clustering** has multiple nodes running applications, monitoring each other
  - Some clusters are for **high-performance computing (HPC)**
    - Applications must be written to use **parallelization**
  - Some have **distributed lock manager** (**DLM**) to avoid conflicting operations

User Mode vs. Kernel Mode

User Mode

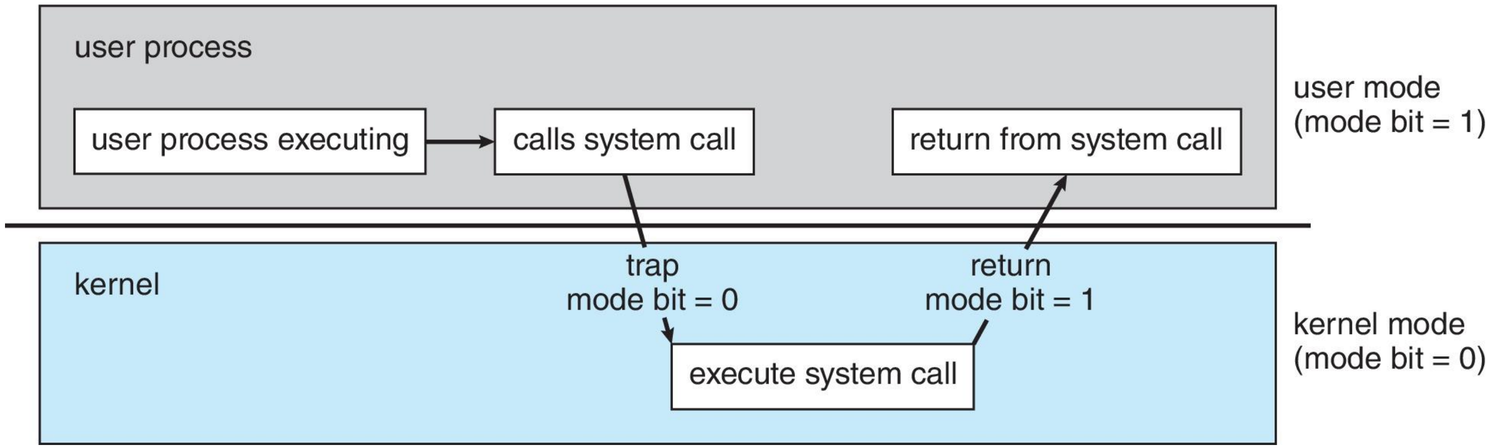
- The system is in user mode when the operating system is running a **user application such as handling a text editor.**
- The transition from user mode to kernel mode occurs when the application requests the help of operating system or an interrupt or a system call occurs.
- The mode bit is set to 1 in the user mode. It is changed from 1 to 0 when switching from user mode to kernel mode.

Kernel Mode

- The **system starts in kernel mode** when it boots and after the operating system is loaded, it executes applications in user mode.
- There are some **privileged** instructions that can only be executed in kernel mode.
- These are interrupt instructions, input output management etc. **If the privileged instructions are executed in user mode**, it is illegal and a trap is generated.
- The mode bit is set to 0 in the kernel mode. It is changed from 0 to 1 when switching from kernel mode to user mode.

Transition from User to Kernel Mode

- Timer to prevent infinite loop / process hogging resources
  - Timer is set to interrupt the computer after some time period
  - Keep a counter that is decremented by the physical clock
  - Operating system set the counter (privileged instruction)
  - When counter zero generate an interrupt
  - Set up before scheduling process to regain control or terminate program that exceeds allotted time



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 termination requires reclaim of any reusable resources

Process Management

- Single-threaded process has one **program counter** specifying location of next instruction to execute
  - Process executes instructions sequentially, one at a time, until completion
- 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

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
- Providing mechanisms for process communication
- Providing mechanisms for deadlock handling

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**
  - 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

File-system Management

- 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)
- 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 directories
    - Mapping files onto secondary storage
    - Backup files onto stable (non-volatile) storage media

Mass-Storage Management

- Usually disks 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
- Entire speed of computer operation hinges on disk subsystem and its algorithms
- **OS activities**
  - Mounting and unmounting
  - Free-space management
  - Storage allocation
  - Disk scheduling
  - Partitioning
  - Protection
- Some storage need not be fast
  - Tertiary storage includes optical storage, magnetic tape
  - Still must be managed – by OS or applications

Caching

- Important principle, performed at **many levels** in a computer (in hardware, operating system, software)
- 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 the 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

Characteristics of Various Types of Storage

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

Movement between levels of storage hierarchy can be explicit or implicit

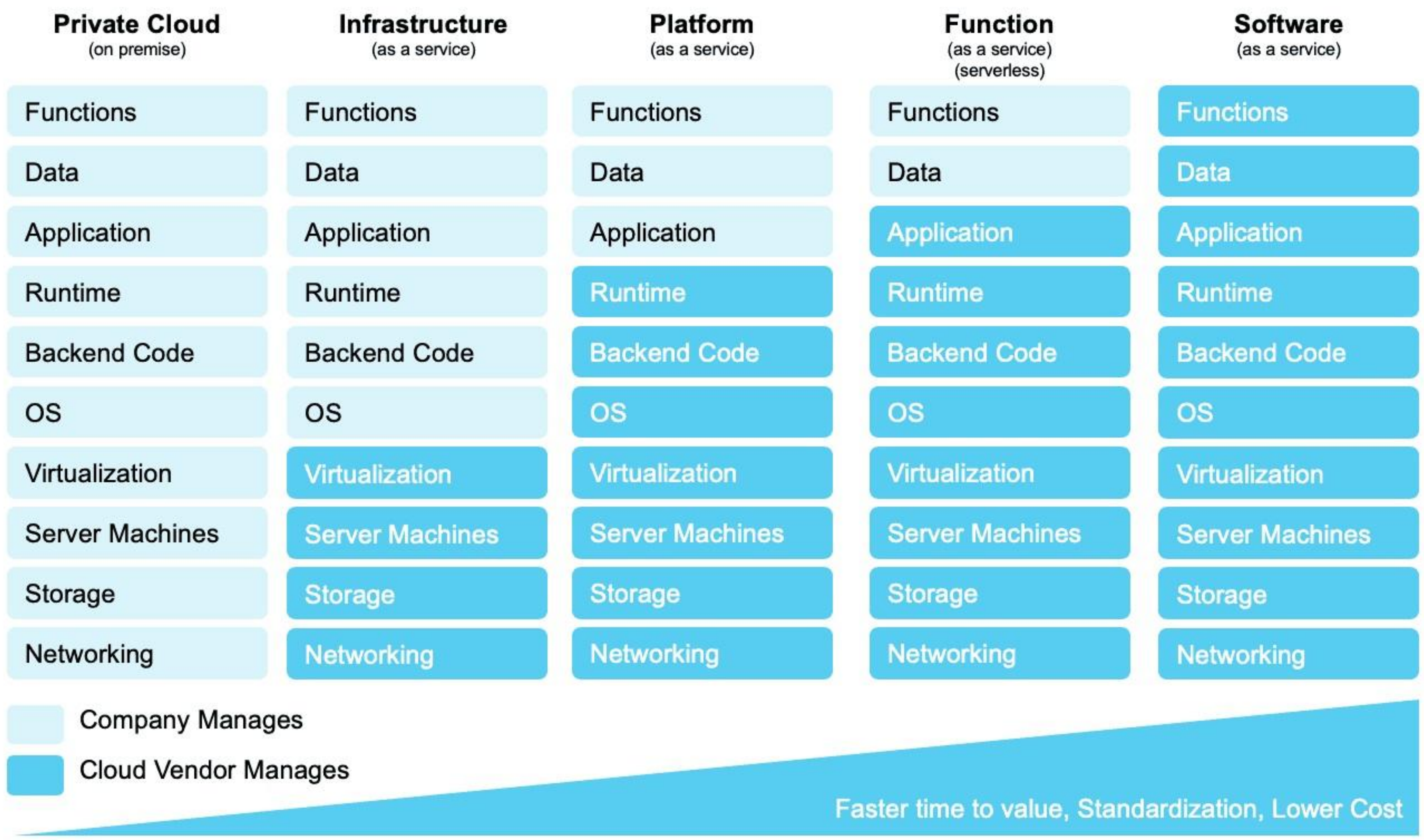
Types of cache memory

- Cache memory is fast and expensive. Traditionally, it is categorized as "levels" that describe its closeness and accessibility to the microprocessor. There are three general cache levels:
- **L1 cache**, or primary cache, is extremely fast but relatively small, and is usually embedded in the processor chip as CPU cache.

- **L2 cache**, or secondary cache, is often more capacious than L1. L2 cache may be embedded on the CPU, or it can be on a separate chip or coprocessor and have a high-speed alternative system bus connecting the cache and CPU. That way it doesn't get slowed by traffic on the main system bus.
- **Level 3 (L3) cache** is specialized memory developed to improve the performance of L1 and L2. L1 or L2 can be significantly faster than L3, though L3 is usually double the speed of DRAM. With multicore processors, each core can have dedicated L1 and L2 cache, but they can share an L3 cache. If an L3 cache references an instruction, it is usually elevated to a higher level of cache.

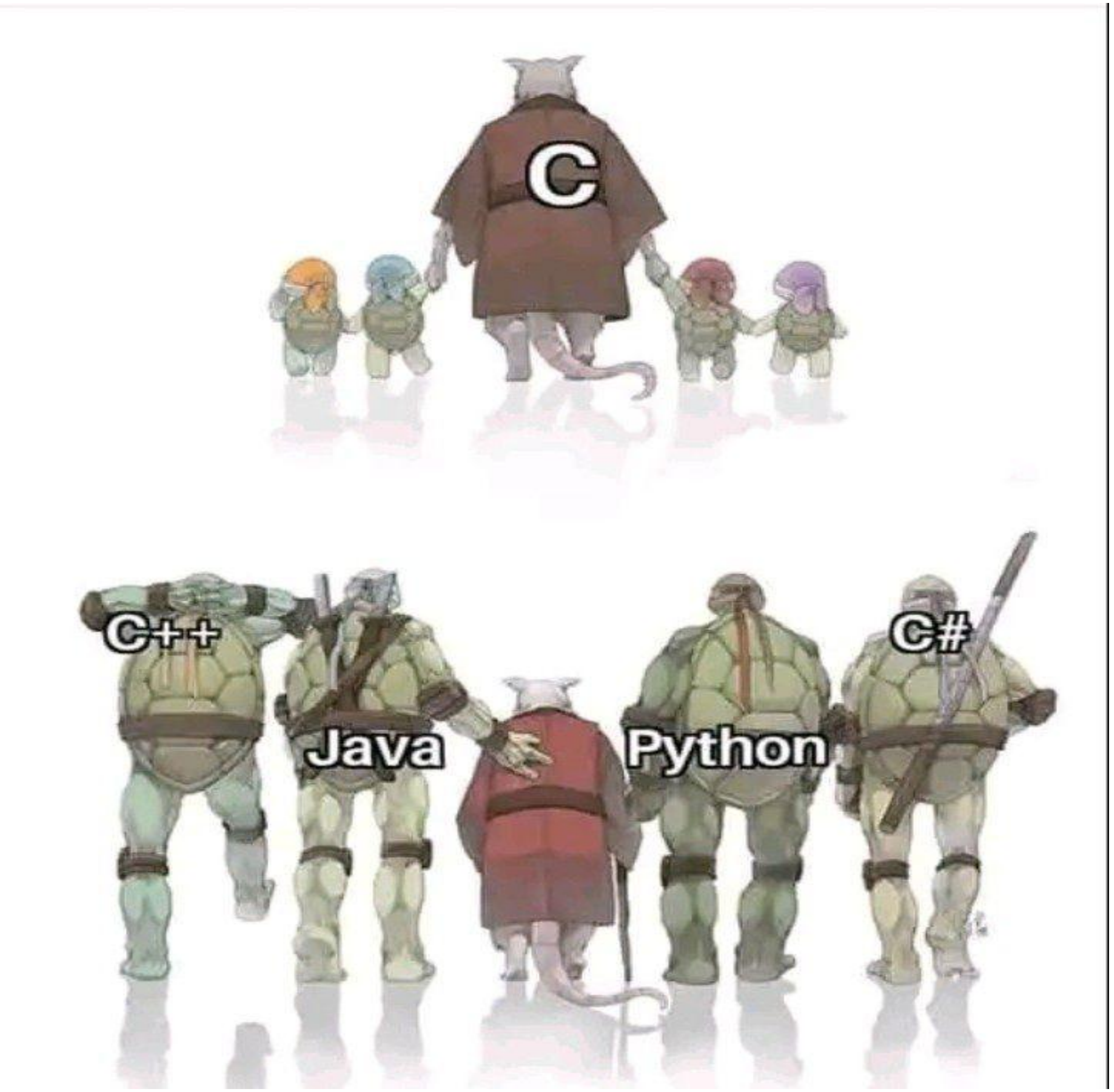
Computing Environments – Cloud Computing

- Delivers computing, storage, even apps as a service across a network
- Logical extension of virtualization because it uses virtualization as the base for it functionality.
  - Amazon **EC2** has thousands of servers, millions of virtual machines, petabytes of storage available across the Internet, pay based on usage
- Many types
  - **Public cloud** – available via Internet to anyone willing to pay
  - **Private cloud** – run by a company for the company's own use
  - **Hybrid cloud** – includes both public and private cloud components
  - Software as a Service (**SaaS**) – one or more applications available via the Internet (i.e., word processor)
  - Platform as a Service (**PaaS**) – software stack ready for application use via the Internet (i.e., a database server)
  - Infrastructure as a Service (**IaaS**) – servers or storage available over Internet (i.e., storage available for backup use)
  - OpenStack OS for the Cloud


Private Cloud
(on premise)
Infrastructure
(as a service)
Platform
(as a service)
Function
(as a service)
(serverless)
Software
(as a service)
Functions
Data
Application
Runtime
Backend Code
OS
Virtualization
Server Machines
Storage
Networking
Company Manages
Cloud Vendor Manages
Faster time to value, Standardization, Lower Cost

Operating-System Structures

Programming Languages of OS


C
C++
Java
Python
C#

HOW OLD IS YOUR LANGUAGE ?

C	C++	C#	Javascript	NodeJs	TypeScript
1972	1980	2001	1995	2009	2012
Objective-C	Swift	Java	Kotlin	Scala	Clojure
1986	2015	1995	2011	2003	2007
Haskell	PHP	Python	Ruby	Erlang	Rust
1990	1995	1990	1995	1986	2010
Go	HTML	CSS	Lisp	Perl	
2009	1991	1996	1958	1987	