DTS207TC Database Development and Design

Lecture 8

More UML

Chap 12. Physical Storage System*

Di Zhang, Autumn 2025

*Titles with * will not be assessed*

Outline

- UML Class Diagram (supplement based on Lec6)
- Structure of a computer*
- Storage System*
- RAID*

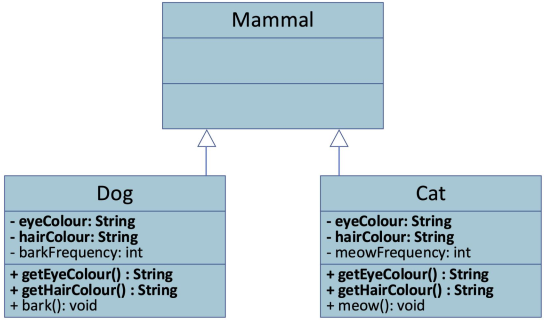
Review of UML class diagram

- Class diagram
  - Structure
  - Relationship
    - Association
    - Dependency
    - Aggregation
    - Composition

Inheritance / Generalization

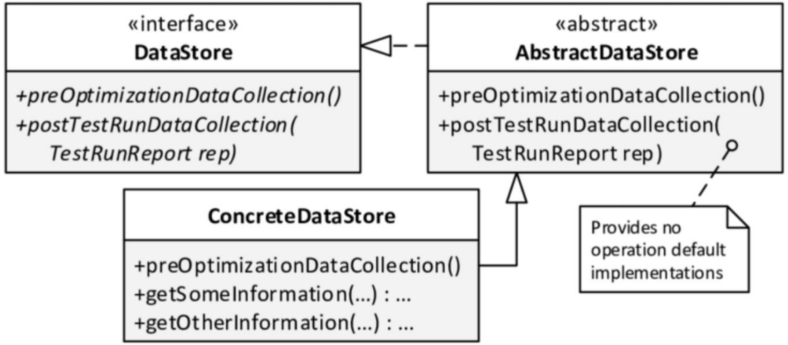
- Content:
  - Definition: Represents an "is-a" relationship between classes.
  - Notation: Hollow triangle + solid line.
  - Example: Animal △ | Dog , Animal △ | Cat.
- Semantics
  - A subclass inherits attributes and operations from its superclass and can extend or override them.
- Notes
  - This is a core concept of OOP, used to achieve polymorphism and code reuse.

1



Interfaces/Abstract class

- Definition:
  - An abstract class is a class that cannot be instantiated and may contain both abstract methods (without implementation) and concrete methods (with implementation).
  - An interface defines a contract that implementing classes must follow. It contains only method signatures (no implementation) and sometimes constants.
- A class implements an interface or abstract class, promising to provide all operations defined in the interface.
- Notation: Hollow triangle + dashed line.
- Example:

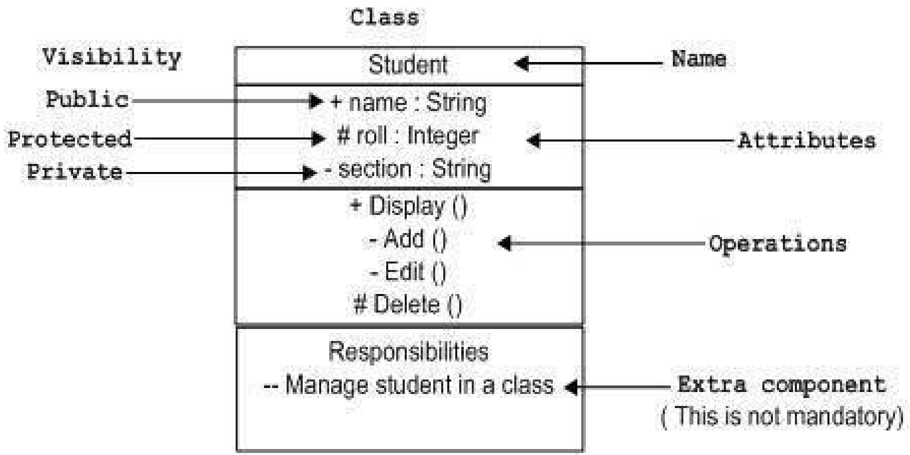


2

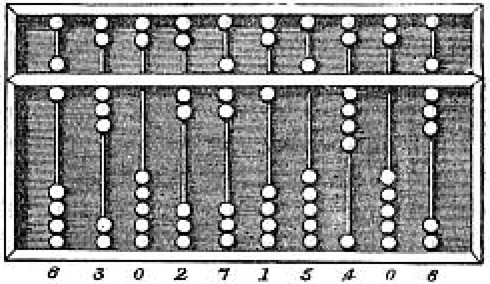
Visibility: The Key to Encapsulation

- + Public: Visible to all classes.
- - Private: Visible only to the owning class.
- # Protected: Visible to the owning class and its subclasses.

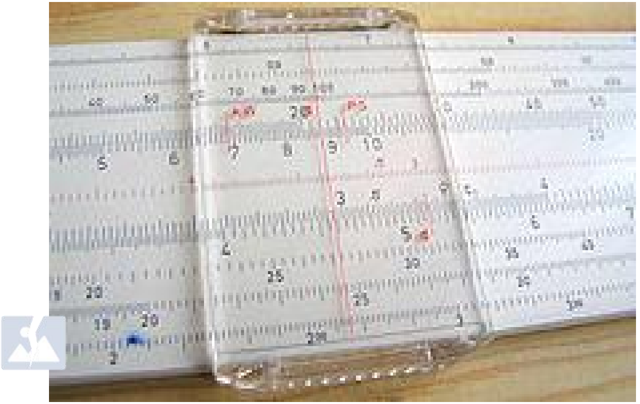
2



What is a Computer?



Suanpan



Slidable Ruler

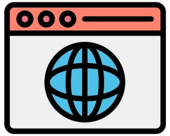
- [Paper Computer]()
- [Three-Body]()
- [Minecraft]()

Calculator

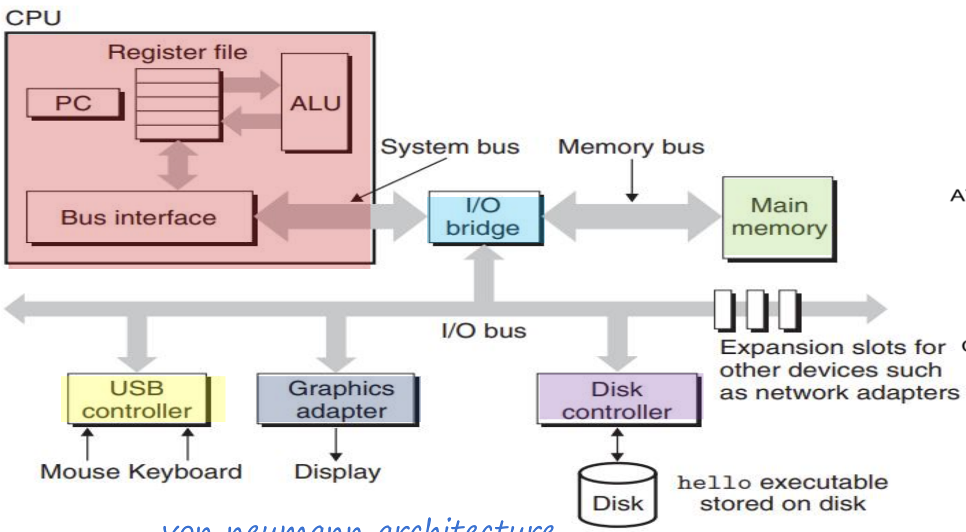
Which of the following are computers?

Computer

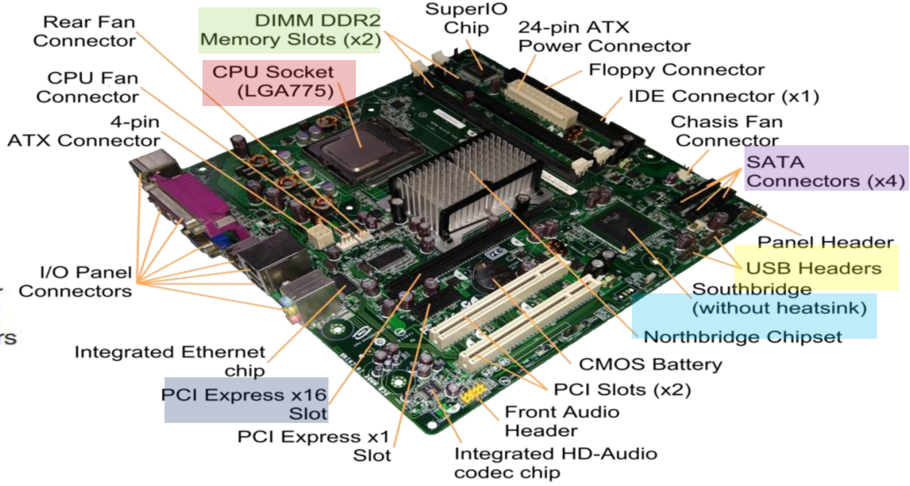
- A computer is a machine that can be **programmed** to automatically carry out sequences of arithmetic or logical operations.
- https://en.wikipedia.org/wiki/Computer



Structure of a personal computer



von neumann architecture

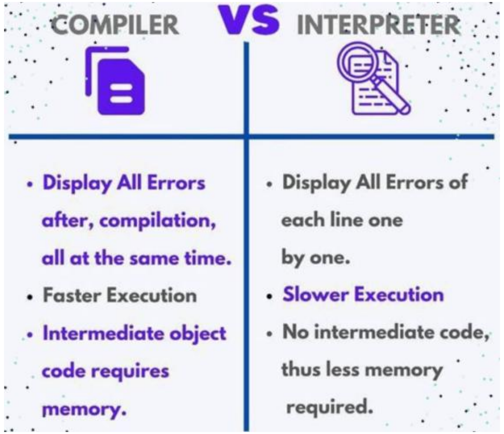


Intel D945GCCR

- The main components (review of 103) :
  - CPU
  - I/O
  - Memory
  - Graphic card

Doom: PC Graphics Benchmark

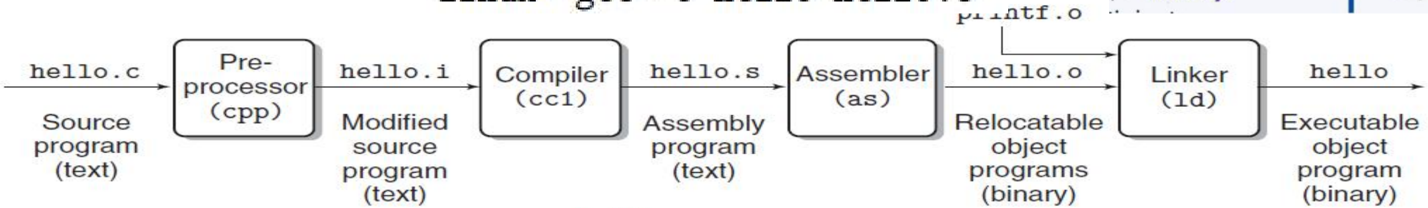
Example: Hello world




```c
#include <stdio.h>

int main()
{
    printf("hello, world\n");
}
```

linux> *gcc -o hello hello.c*

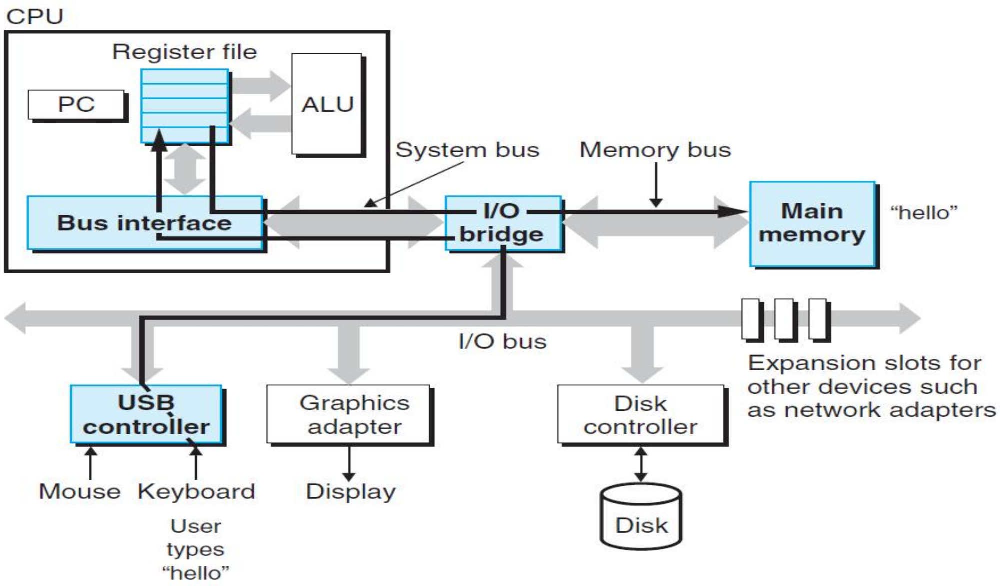


Dealing with #include

```
main:
    subq    $8, %rsp
    movl    $.LC0, %edi
    call    puts
    movl    $0, %eax
    addq    $8, %rsp
    ret
```

Q: Anyone knows differences between C and Python?

Example: Hello world

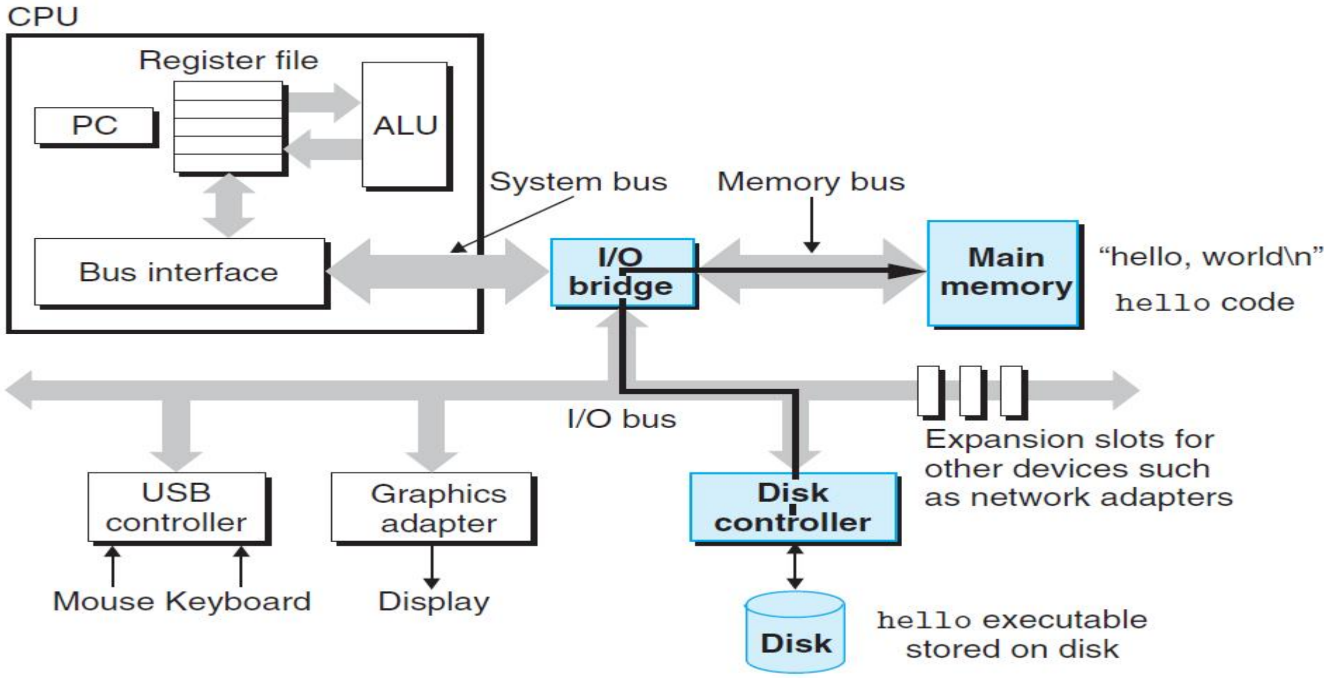



```
unix> ./hello
hello, world
unix>
```

1. Shell reads the hello command into memory from the keyboard.

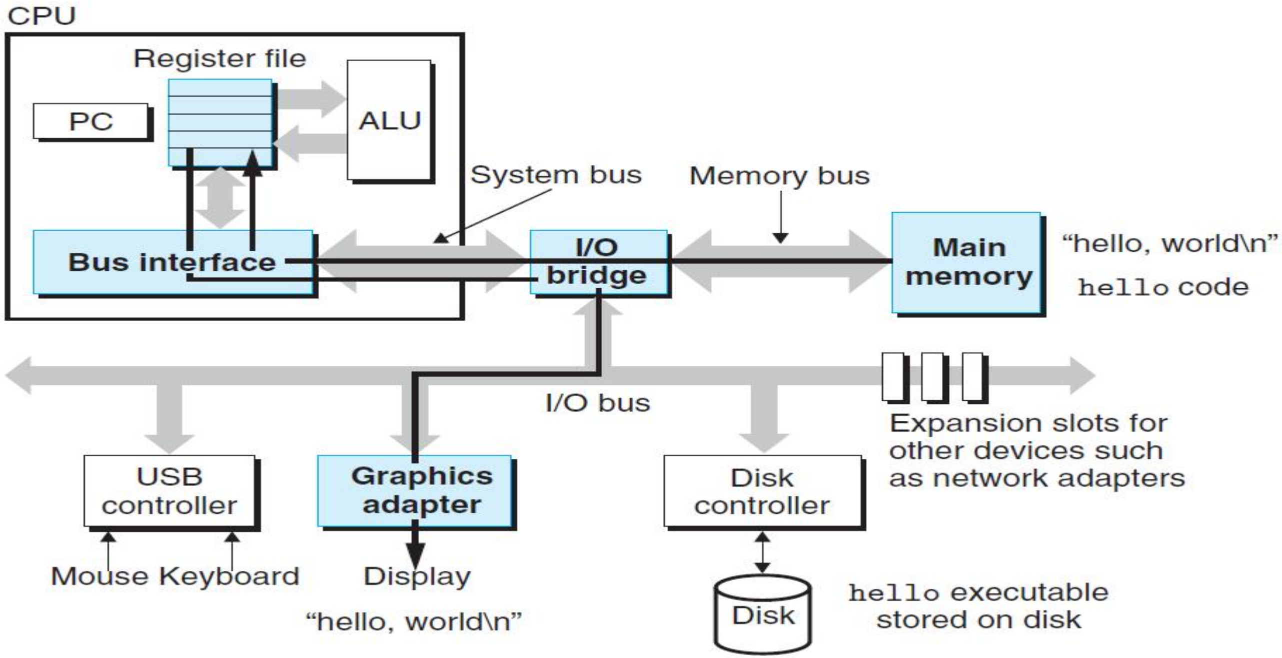
Example: Hello world

Xi'an Jiaotong-Liverpool University 西交利物浦大学



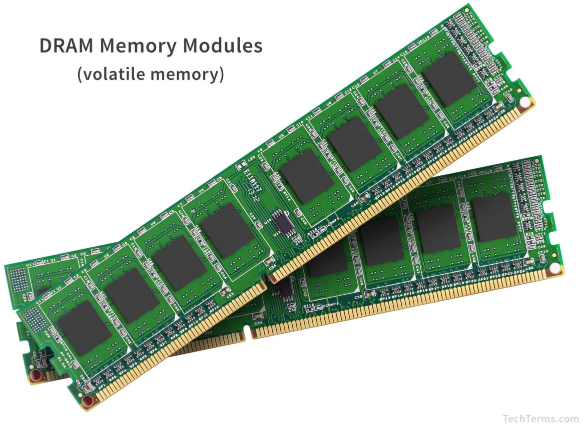
2. OS loads the executable from disk into main memory.

Example: Hello world



3. The OS prepares the program, which then executes itself and writes the output string from memory to the display.

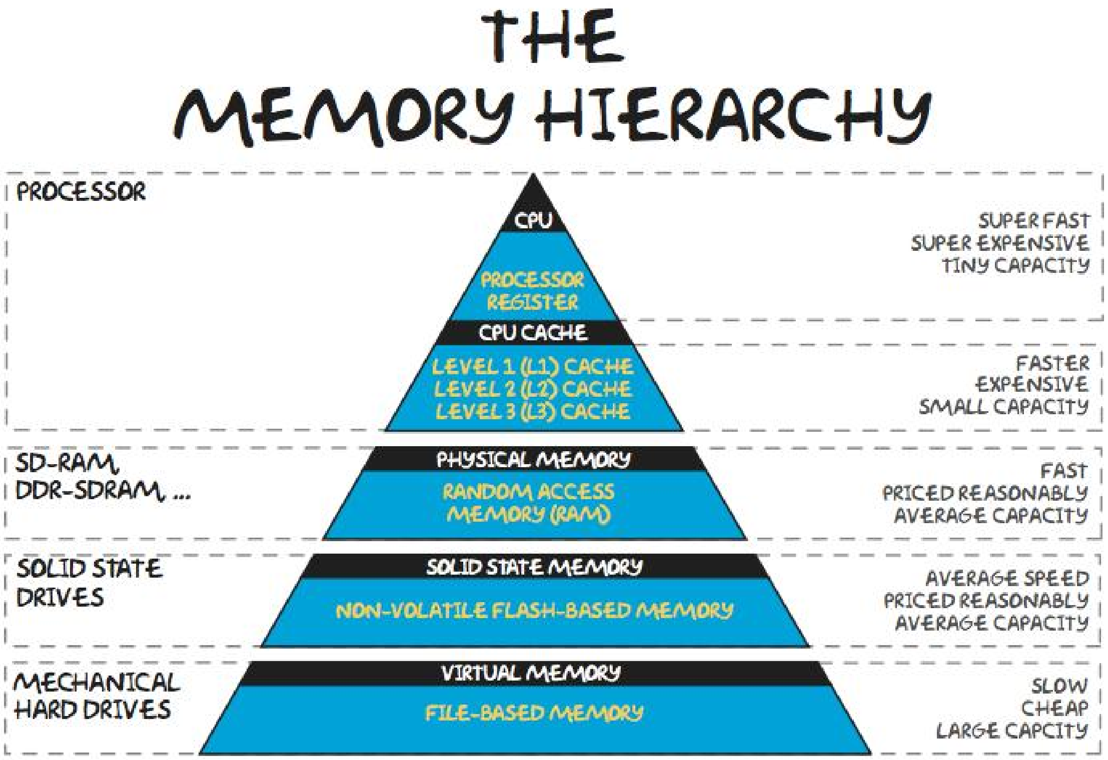
Classification of Physical Storage Media



- Can differentiate storage into:
  - **volatile storage:** loses contents when power is switched off
  - **non-volatile storage**:
    - Contents persist even when power is switched off.
    - Includes secondary and tertiary storage, as well as batter-backed up main-memory.
- Factors affecting choice of storage media include
  - Speed with which data can be accessed
  - Cost per unit of data
  - Reliability

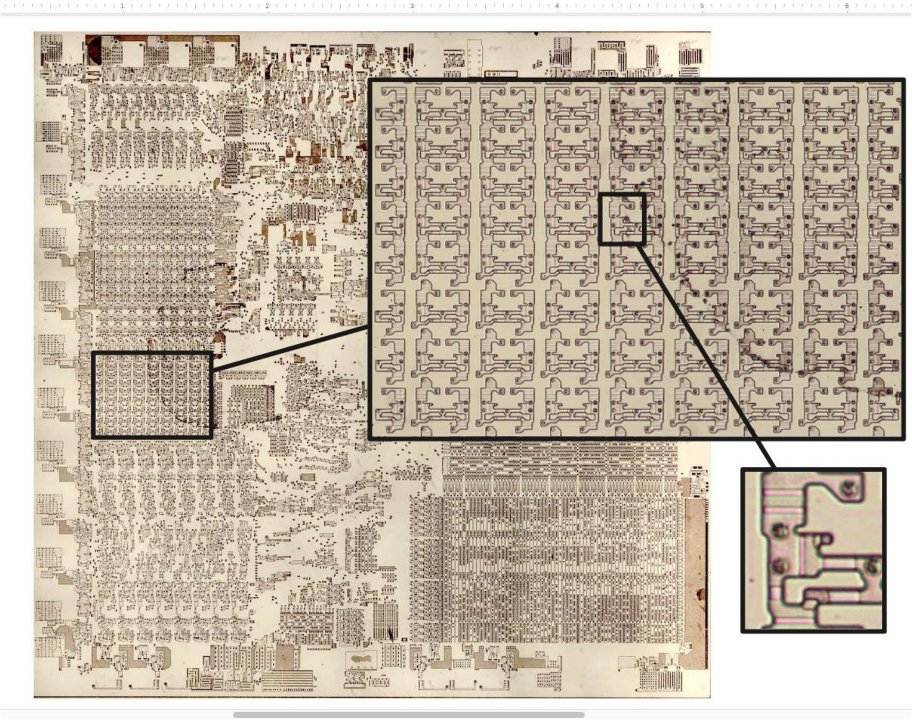
Memory Hierarchy

- What is the Memory Hierarchy?
  - A trade-off between Speed, Cost, and Capacity.
  - No single storage technology is optimal for all three.
  - The solution: A tiered system where data is moved between layers to create the illusion of a large, fast, and cheap memory.
- Key Principles:
  - Locality of Reference:
    - Temporal Locality: Recently accessed data is likely to be accessed again soon.
    - Spatial Locality: Data near recently accessed data is likely to be accessed soon.
  - Caching: Storing copies of frequently used data in faster storage.



CPU Register

- Location: Inside the CPU core.
- Function: Hold the data the CPU is currently processing (e.g., operands for arithmetic operations, memory addresses).
- Key Characteristics:
  - Speed: Fastest memory, operates at CPU clock speed (nanoseconds).
  - Size: Smallest capacity (typically a few hundred to a few thousand bytes).
  - Volatility: Volatile (loses data without power).
  - Cost: Most expensive per bit.
  - Managed by: The compiler and the CPU hardware.

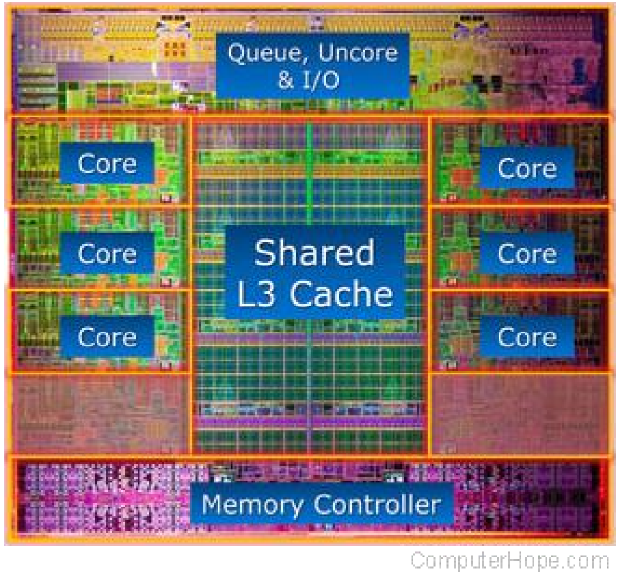


The Intel 8086 processor's registers

Cache

- Location: On the CPU chip (L1, L2) or nearby (L3).
  - Function: A buffer between the CPU and main memory. It holds copies of frequently used data from main memory to reduce access latency.
- Layers (L1, L2, L3):
  - L1: Split into Instruction and Data cache. Smallest and fastest, per CPU core.
  - L2: Larger than L1, slower, often per core.
  - L3: Shared among all cores. Largest on-chip cache, slowest of the three.
- Key Characteristics:
  - Speed: Very fast (10s of nanoseconds).
  - Size: Typically 256KB - 64MB total.
  - Volatility: Volatile.
  - Managed by: Hardware automatically.

Intel Core i7-3960X Processor Die Detail

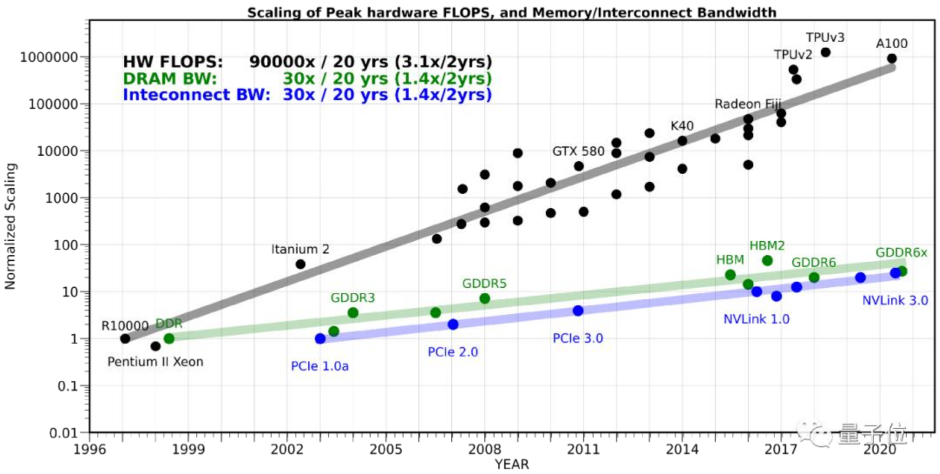


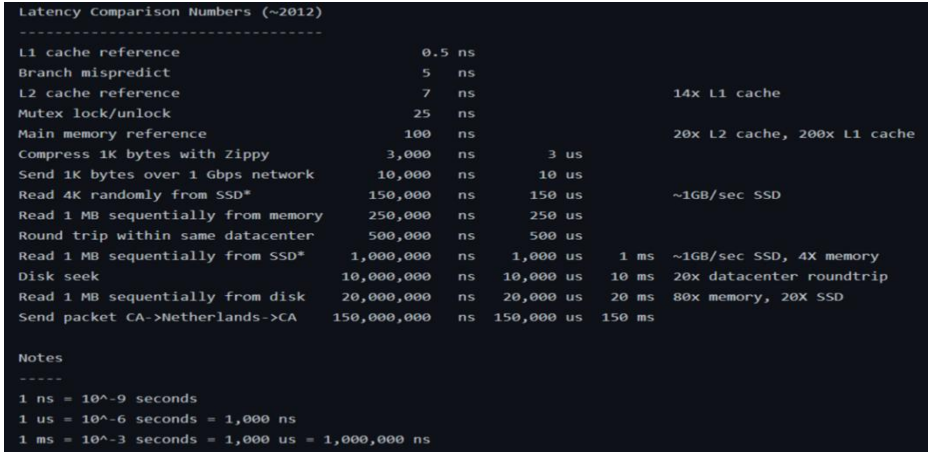
Main Memory

- Location: On the motherboard, connected to the CPU via a memory bus.
- Function: Holds all the data and instructions that the CPU is actively using or may use shortly. It is the "working area" for the operating system and applications.
- Types:
  - DRAM (Dynamic RAM): The standard. Needs constant refreshing. Higher density, lower cost.
  - SRAM (Static RAM): Faster, no refresh needed. Used for Cache (more expensive).
- Key Characteristics:
  - Speed: Slower than cache (~100 nanoseconds).
  - Size: Much larger than cache (typically 8GB - 256GB in modern systems).
  - Volatility: Volatile (this is why you lose unsaved work on a power cut).
  - Cost: Cheaper than cache, more expensive than storage.

Speed Mismatch: Memory Wall

Xi'an Jiaotong-Liverpool University
西交利物浦大学





```
Latency Comparison Numbers (~2012)
----------------------------------
L1 cache reference                           0.5 ns
Branch mispredict                            5   ns
L2 cache reference                           7   ns                      14x L1 cache
Mutex lock/unlock                           25   ns
Main memory reference                      100   ns                      20x L2 cache, 200x L1 cache
Compress 1K bytes with Zippy             3,000   ns        3 us
Send 1K bytes over 1 Gbps network       10,000   ns       10 us
Read 4K randomly from SSD*             150,000   ns      150 us          ~1GB/sec SSD
Read 1 MB sequentially from memory     250,000   ns      250 us
Round trip within same datacenter      500,000   ns      500 us
Read 1 MB sequentially from SSD*     1,000,000   ns    1,000 us    1 ms  ~1GB/sec SSD, 4X memory
Disk seek                           10,000,000   ns   10,000 us   10 ms  20x datacenter roundtrip
Read 1 MB sequentially from disk    20,000,000   ns   20,000 us   20 ms  80x memory, 20X SSD
Send packet CA->Netherlands->CA    150,000,000   ns  150,000 us  150 ms

Notes
-----
1 ns = 10^-9 seconds
1 us = 10^-6 seconds = 1,000 ns
1 ms = 10^-3 seconds = 1,000 us = 1,000,000 ns
```

GFLOPS=Giga (10^9) Floating-point Operations Per Second

- Problem: In order to read and write data, the CPU must wait for the memory

An Example of Speed Mismatching

Flash Memory

- Location: Inside the computer (M.2, SATA) or as an external device.
- Function: The standard for persistent, high-speed storage. Replacing magnetic disks as the primary boot and storage drive.
- Technology: Non-volatile. Uses NAND gates to trap electrons, representing 0s and 1s.
- Key Characteristics:
  - Speed: Much faster than HDDs (microseconds for access), but slower than RAM.
  - Size: Large capacity (256GB - several Terabytes).
  - Volatility: Non-Volatile.
  - Cost: More expensive per GB than HDDs, but cheaper than RAM.
  - Drawbacks: Limited write cycles (wear leveling is used to mitigate this).

Magnetic-disk storage

- Location: Inside the computer or external enclosures.
- Function: Traditional mass storage for data that is not frequently accessed. Cost-effective for large capacities.
- Technology: Uses rotating magnetic platters and a read/write head that moves across the surface.
- Key Characteristics:
  - Speed: Slow (milliseconds for access due to mechanical seek time and rotational latency).
  - Size: Very large capacity (up to 20TB+).
  - Volatility: Non-Volatile.
  - Cost: Cheapest per GB among active storage devices.
  - Drawbacks: Mechanical parts are sensitive to shock and wear.

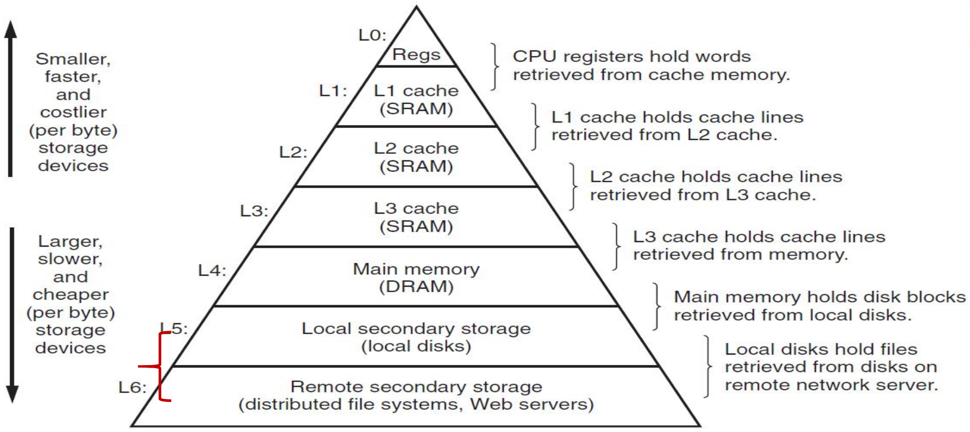
Optical Storage (CD, DVD, Blu-ray)

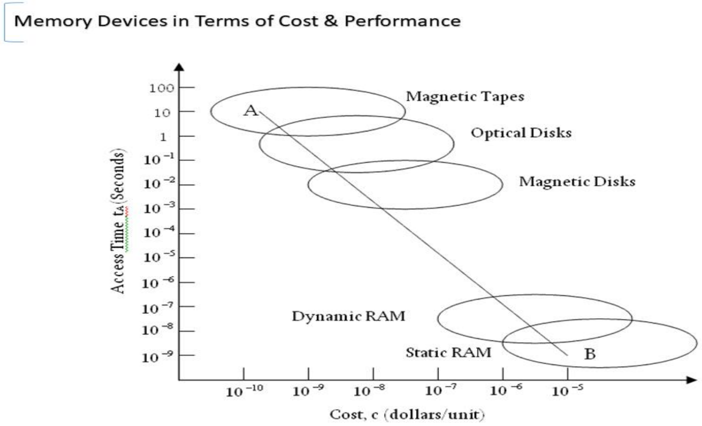
- Technology: Uses lasers to read/write pits and lands on a reflective disc.
- Use Case: Software distribution, media (movies, music), personal backups.
- Characteristics: Very slow, portable, low capacity (700MB - 100GB), very cheap per disc.

Magnetic Tape

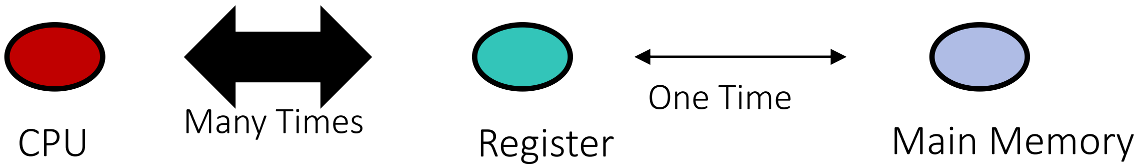
Floopy disk

Caches matter





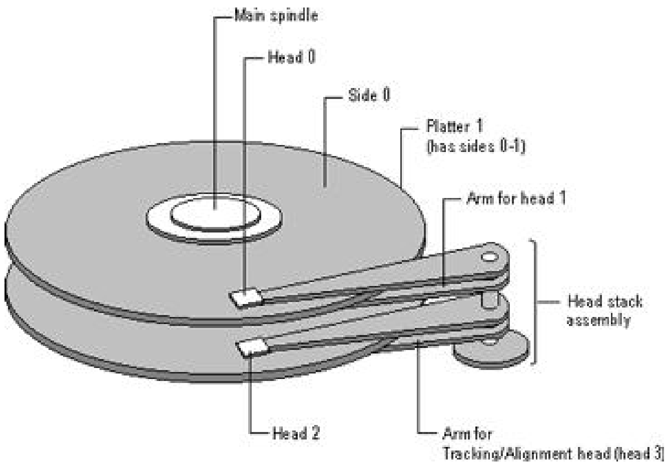
They can be cache at bottom!



- Register: fast but expensive
- The role of cache: play the role of agent; based on the principle of locality, it can speed up the visiting
- Hierarchical cache system: Multi-Level Acceleration

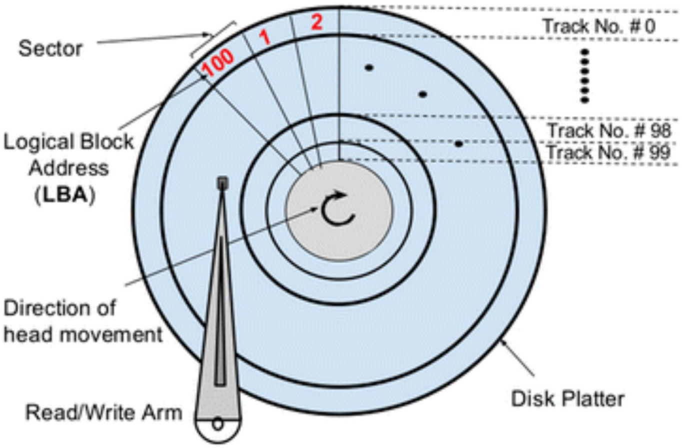
Hard Disk Drives

- Key Physical Components:
  - Platters: Rigid, circular disks coated with a magnetic material. Data is stored here.
  - Spindle: The central axle that rotates the platters at high speed (e.g., 5400, 7200, 15000 RPM).
  - Read/Write Heads: Tiny electromagnets that read from and write to the platter surfaces. There's one head per platter side.
  - Actuator Arm: The mechanical arm that holds the read/write heads and moves them across the platters.
  - Actuator: The motor that precisely positions the actuator arm.



Data Organization

- Data Organization:
  - Tracks: Concentric circles on a platter.
  - Sectors: Arc-shaped segments of a track (typically 512 bytes or 4KB).
  - Cylinder: The set of all tracks at the same actuator arm position across all platters.

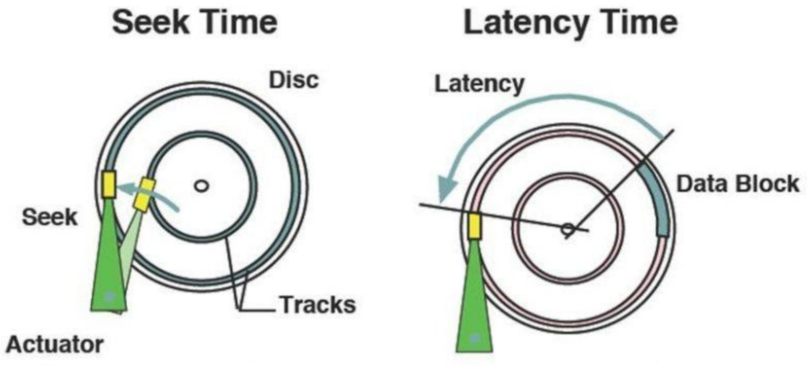


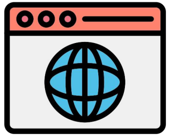
Geometry of the HDD with 1 platter, 100 sectors/track and 100 tracks/platter

Key Performance Metrics

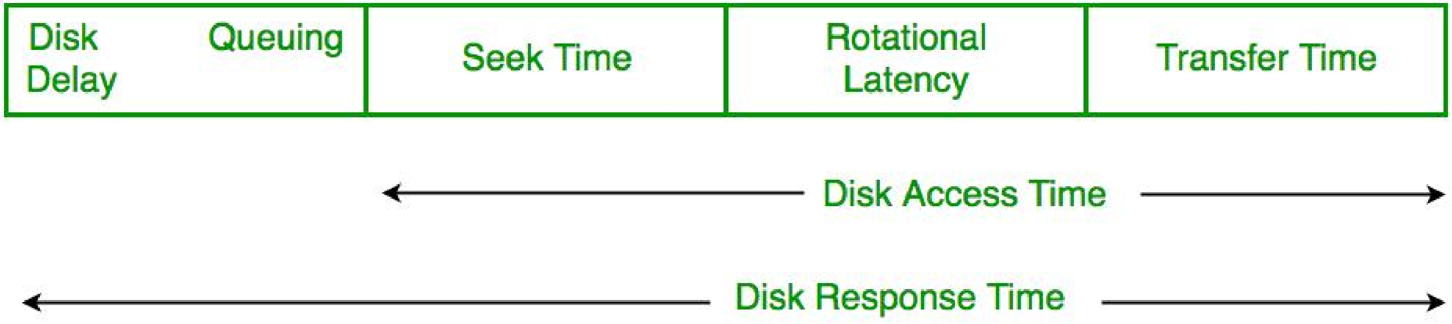
Xi'an Jiaotong-Liverpool University 西交利物浦大学

- Performance is dominated by mechanical movement. We measure three key things:
- 1. Seek Time:
  - What it is: The time for the actuator arm to move the heads to the correct track.
  - Analogy: Moving the needle on a record player to the right song.
  - Typical Value: A few milliseconds (ms).
- 2. Rotational Latency:
  - What it is: The time for the desired sector to rotate under the read/write head after the arm is in position.
  - Calculation: On average, it's half the time for one full rotation. For a 7200 RPM disk: (60 sec/min) / (7200 rotations/min) / 2 ≈ 4.17ms.
- 3. Data Transfer Rate:
  - What it is: The speed at which data is read from or written to the platter once the head is positioned.
  - Factors: Determined by platter areal density and rotational speed.





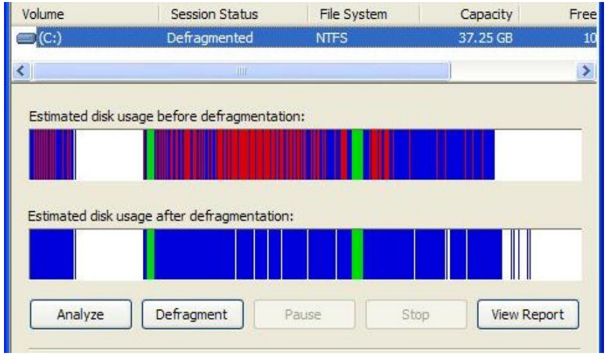
The Latency Bottleneck & Total Access Time



- Total Average Access Time:
  - The sum of Seek Time, Rotational Latency, and a small overhead for data transfer.
  - Formula: Access Time ≈ Seek Time + Rotational Latency
- The Mechanical Bottleneck:
  - This sum is why HDDs are slow compared to SSDs (which have near-zero seek/rotation times).
  - Access times are typically in the range of 5-15 ms.
  - Fragmentation makes this worse, forcing the head to seek all over the disk for a single file.
- IOPS (Input/Output Operations Per Second):
  - A common performance benchmark derived from access time.
  - Rough Calculation: IOPS ≈ 1 / (Access Time in seconds). A disk with 10ms access time can theoretically handle ~100 IOPS.

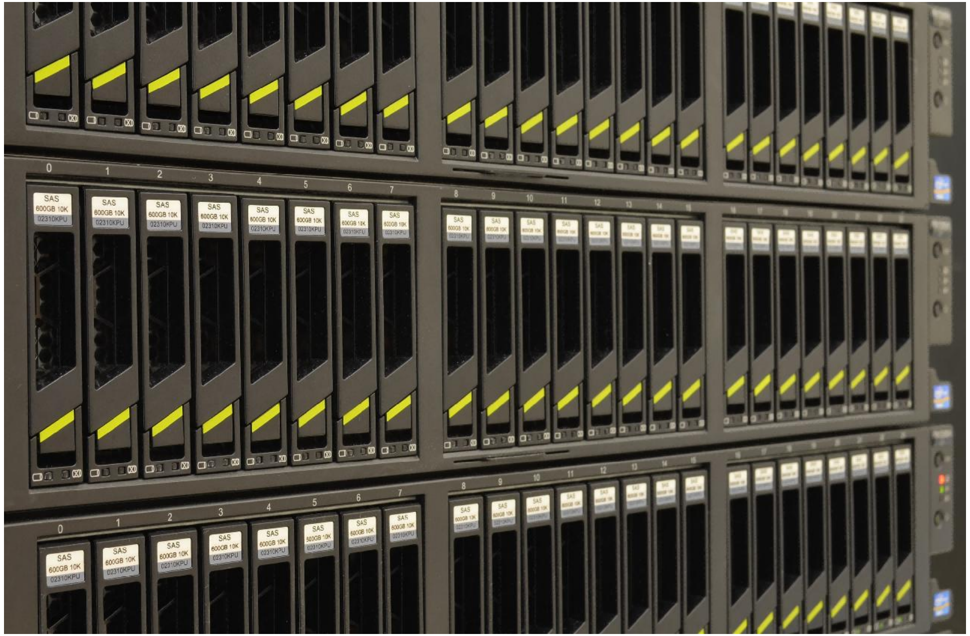
Real-World Impact & Trends

- How This Affects You:
  - Boot Time/App Loading: Involves reading thousands of small files from random locations (high seek time).
  - Large File Operations: (e.g., video editing) are faster as they involve sequential reads (minimizing seek).
- The Shift to SSDs:
  - SSDs have no moving parts, eliminating seek time and rotational latency.
  - This results in access times ~100x faster than HDDs.
- HDD's Enduring Role:
  - HDDs remain dominant for cold storage and massive capacities (e.g., data centers, backup archives) due to their lower cost per Gigabyte.
- Conclusion: Understanding HDD mechanics explains the fundamental performance characteristics that have shaped computing for decades.



What is RAID?

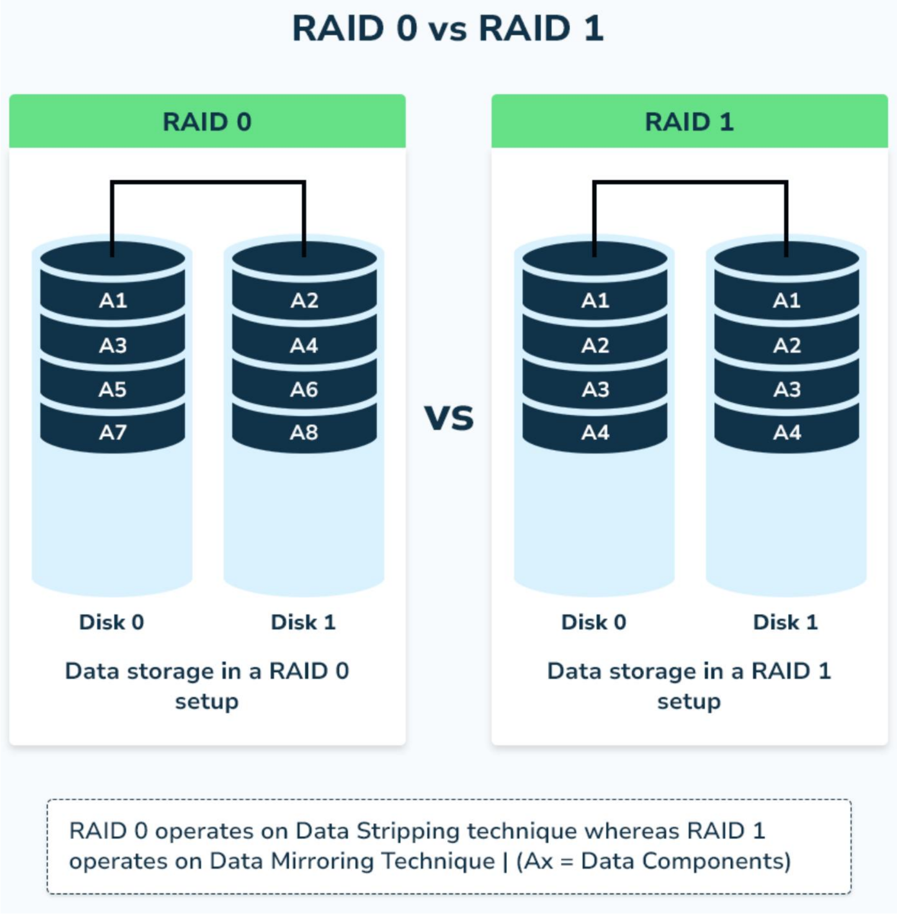
- RAID stands for Redundant Array of Independent Disks.
- It's a technology that combines multiple physical hard drives into one logical unit.
- Key Goals:
  - Increase Performance (Data is read/written to multiple disks at once).
  - Provide Redundancy (Data is protected if a drive fails).
  - Create Larger Storage Volumes (Combine smaller drives into one big one).



Common RAID Levels (Part 1)

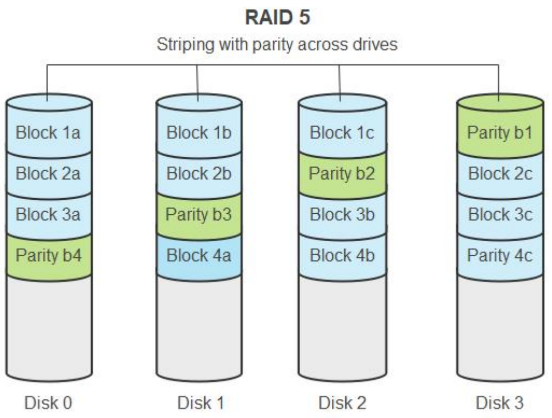
Xi'an Jiaotong-Liverpool University
西交利物浦大学

- RAID 0 (Striping)
  - How it works: Data is split into blocks and spread across all drives.
  - Pros: Excellent Performance (both read and write).
  - Cons: No Redundancy. One drive fails, all data is lost.
  - Use Case: Video editing, gaming—where speed is critical and data is temporary.
- RAID 1 (Mirroring)
  - How it works: Data is written identically to two (or more) drives.
  - Pros: Excellent Redundancy. If one drive fails, no data is lost.
  - Cons: High cost (50% storage efficiency).
  - Use Case: Critical file servers, system drives.

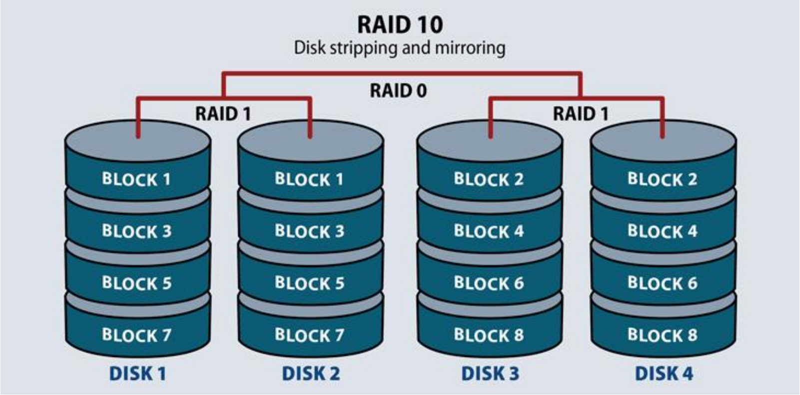


Common RAID Levels (Part 2)

- RAID 5 (Striping with Distributed Parity)
  - How it works: Data and parity information are striped across three or more drives. Parity allows for data reconstruction if one drive fails.
  - Pros: Good read speed, efficient storage (loses only one drive's worth of capacity).
  - Cons: Slow write speeds due to parity calculation. Risky during rebuild.
  - Use Case: General purpose file and application servers.



- RAID 10 (1+0: A Mirror of Stripes)
  - How it works: Combines RAID 1 and RAID 0. Disks are mirrored in pairs, and then the pairs are striped.
  - Pros: High performance and high redundancy. Can survive multiple drive failures (if in the right mirrors).
  - Cons: Highest cost (50% storage efficiency).
  - Use Case: High-performance databases, critical applications.



Lab

- This week's lab is about Rapidminer - a very powerful tool with GUI
- Zero-code
- Please register an eductional accout before lab!