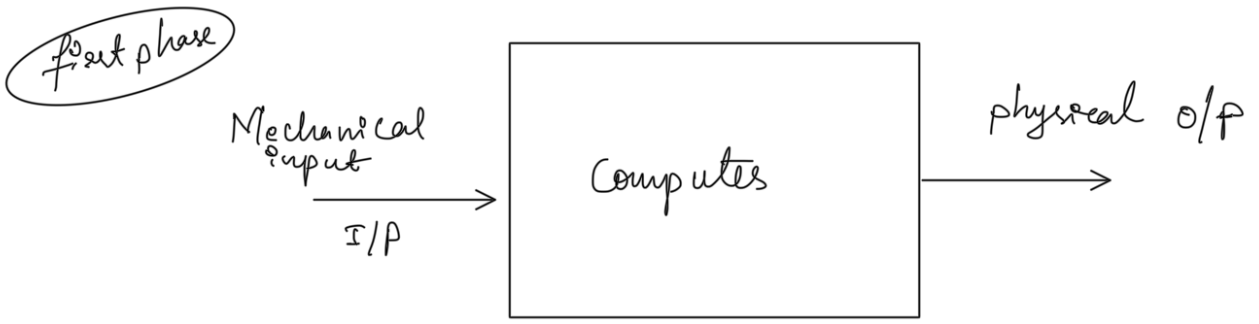
Day 1
11/07/2025

Basics of Computer Science

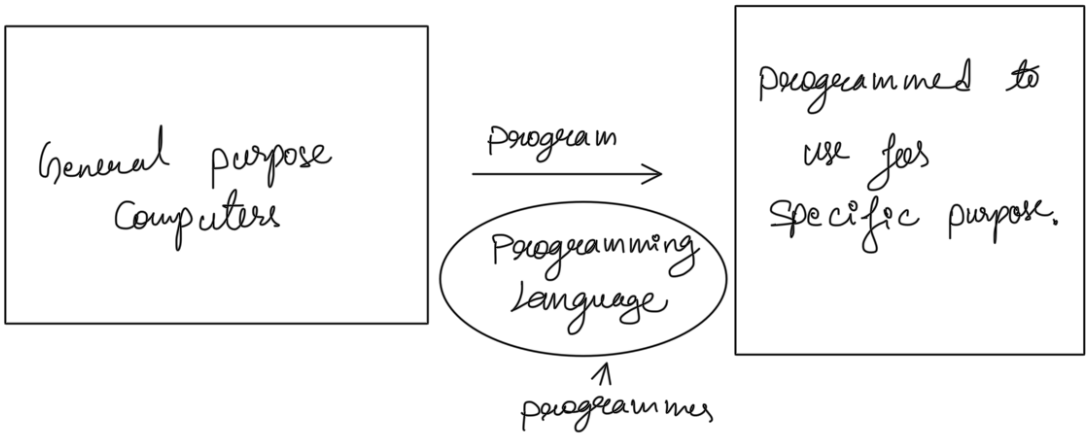
* Evolution of computers



① Initial days computers are large in size
Initial Days its used for High End Calculations like mathematical calculations, research for universities Ent. used for specific High End mathematical calculations.

Exa of general purpose computers

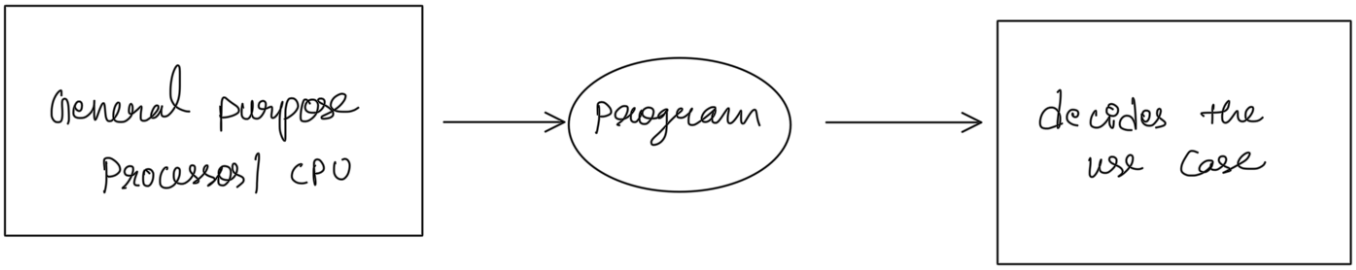
During manufacturing of general purpose computers Initially their objective, purpose & goal are decided

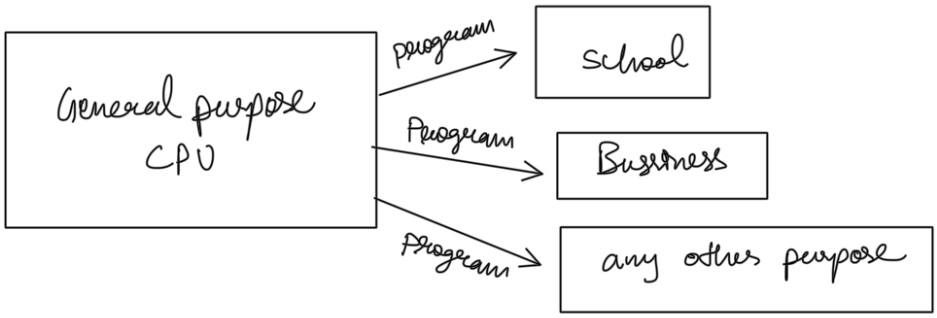


Here where the programming came into picture.

The general purpose computers are programmed to use it for specific tasks. For Ex: for schools, Small Business Ent...

This general purpose computer made Big impact on field of programming and the Programmers Job creation





Micro processors

Evolution

8085 / 8086

Intel x 86

ARM

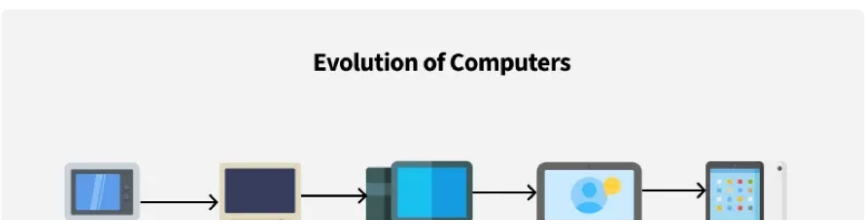
This is How the general purpose Computer's Evolved

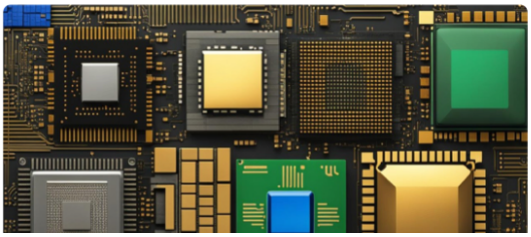
From → only used By Scientists

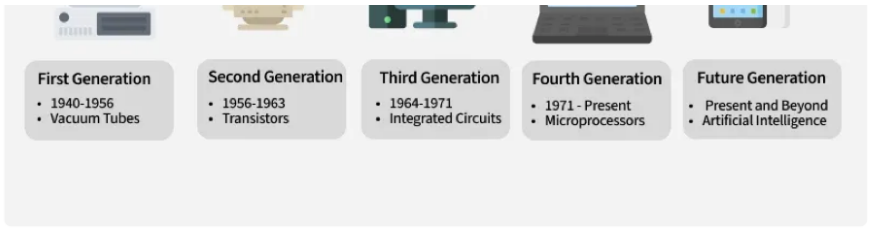
To → used By Every Single person with the help of Programming.

Computer's are moving towards Edge → Each Corner of the Real World.

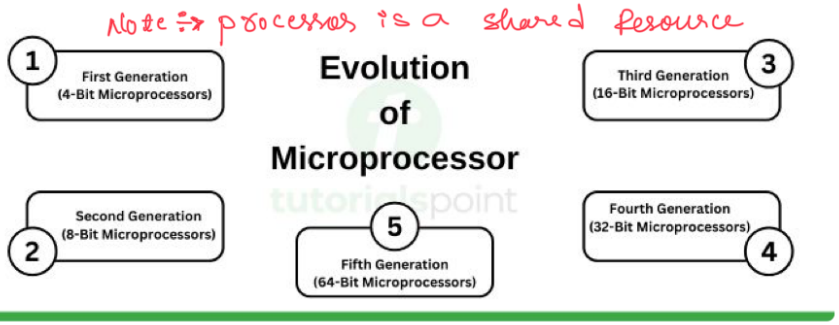
Its still Evolving and Research is going on future generation Computers.

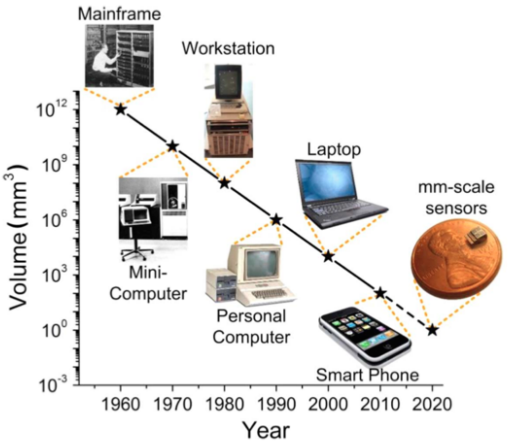






Note :- processor is a shared resource





Processors do all the task of computers the programming makes / guides processors to do the work.

Parts of Computers / Anatomy of computers

1. Mother Board → Connects all others parts of computer
2. processor / CPC
3. Memory (RAM) → Random axis memory → Temprory Memory.
   2, 4, 8, 16, 32 GB
   → 2^x Exponential

Ram Rembers temporary data fetched from SSD / HDD for program Execution.
Ram temperorly stores the data to be saved or fetched from storage (HDD / SDD)

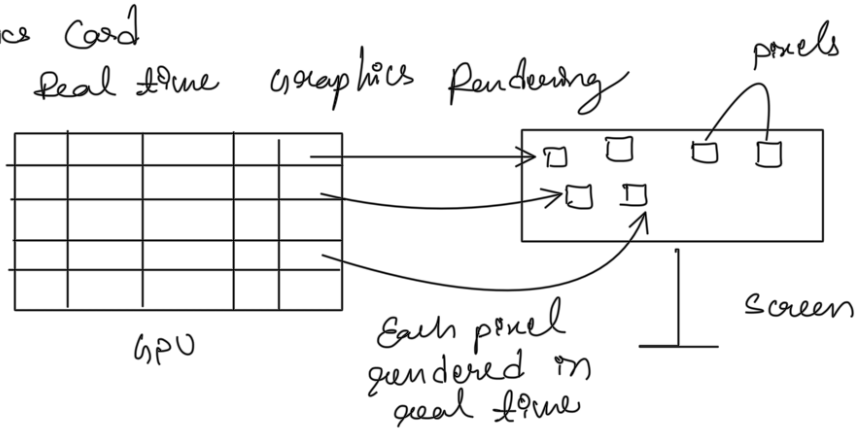
4. HDD / SSD → Permanent Memory (Read & Wrote)
   HDD :- Hard disk drive
   SDD :- Solid state drive

5. Sound Card :- Audio input output Handling
6. Graphics Card :- used in wide Range (High graphics Rendering)
7. power Unit :- Distribute AC → DC current / voltage to the all other connected parts of mother board

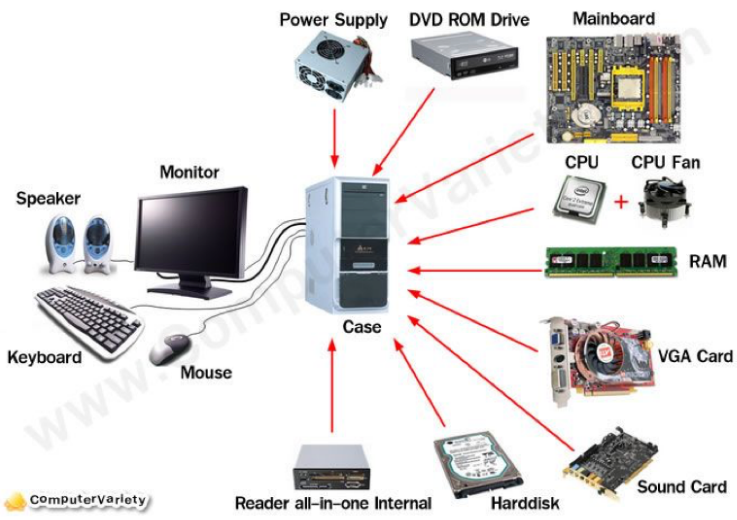
accordingly their use case.

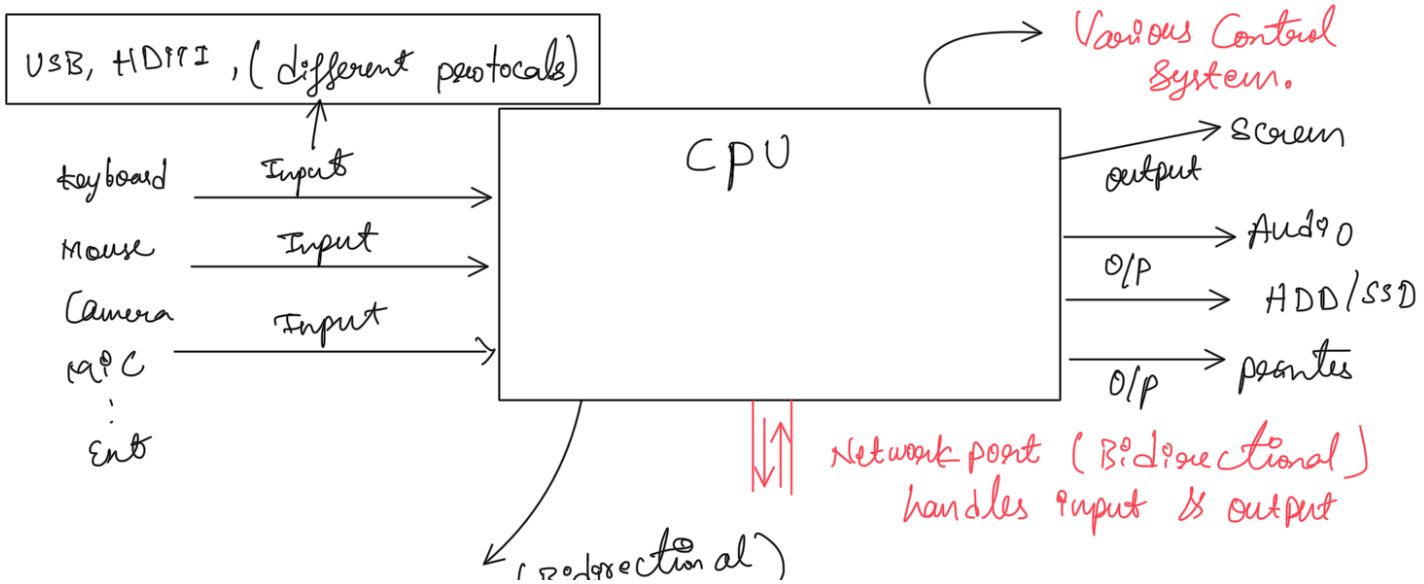
Graphics Card

Real time Graphics Rendering



Also used in Machine Learning for High Data processing parallely data processing is done during Machine Learning Algorithms Execution.





Touch screen (Bidirectional data flow)

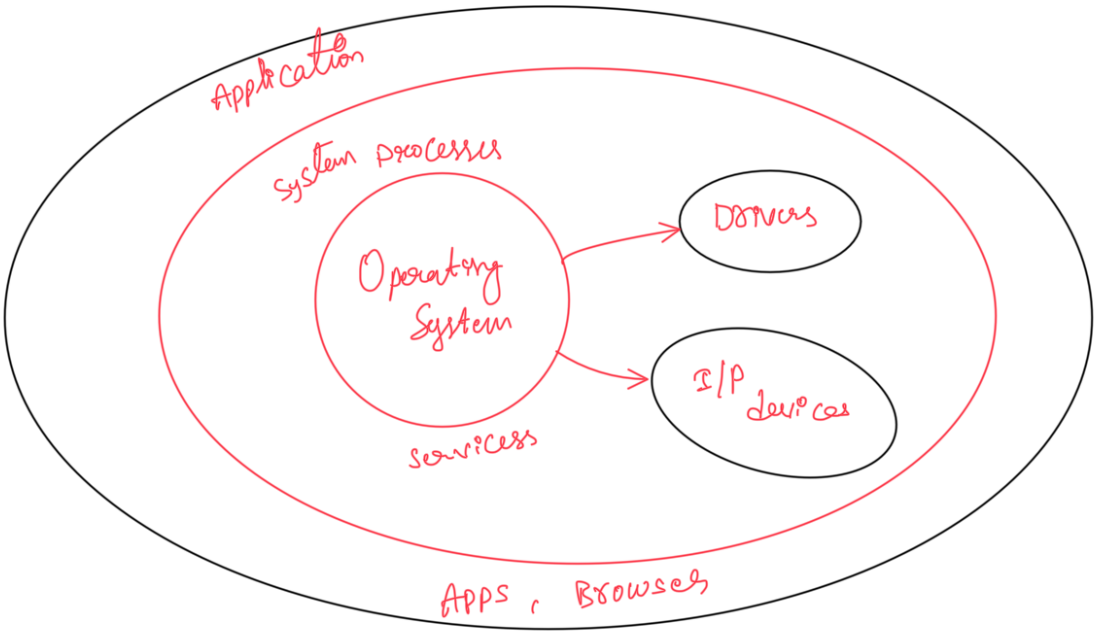
Different protocols used to read / write Data
Protocols depends on various Input / output devices.

Operating System (Basic understanding)

OS → Boss of Computer

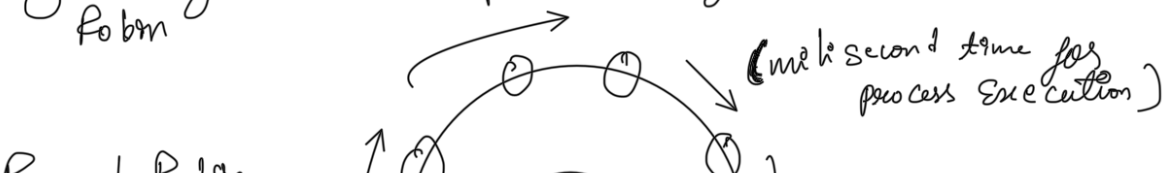
Every task needs permission of OS

Operating System is Core of System.

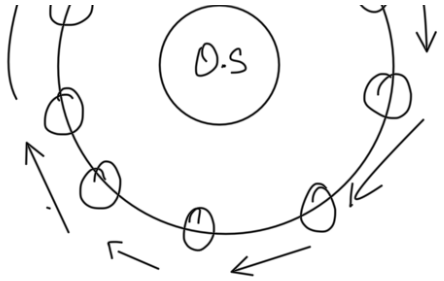


Security & permissions are taken care By OS

Shared Resources like Network, processor, Memory, input/output devices access is controlled By Operating System By using Some special Algorithms like Round Robin



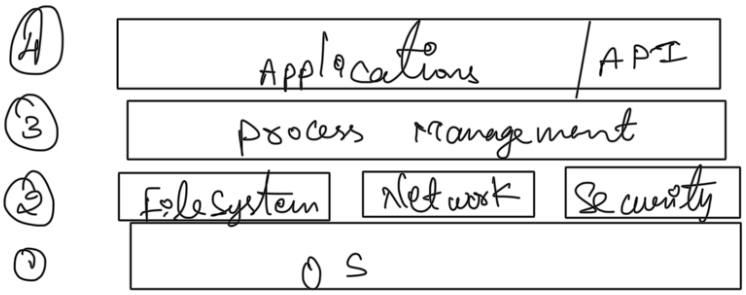
found from



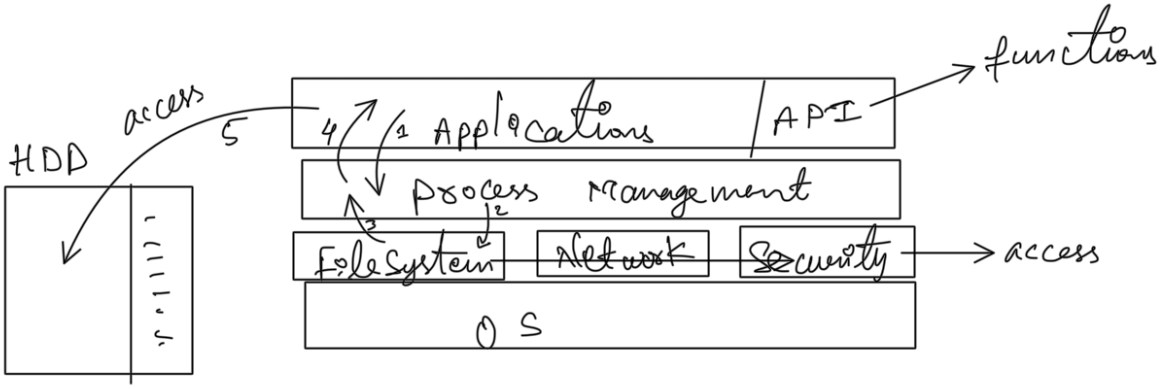
Basic of OS Blocks

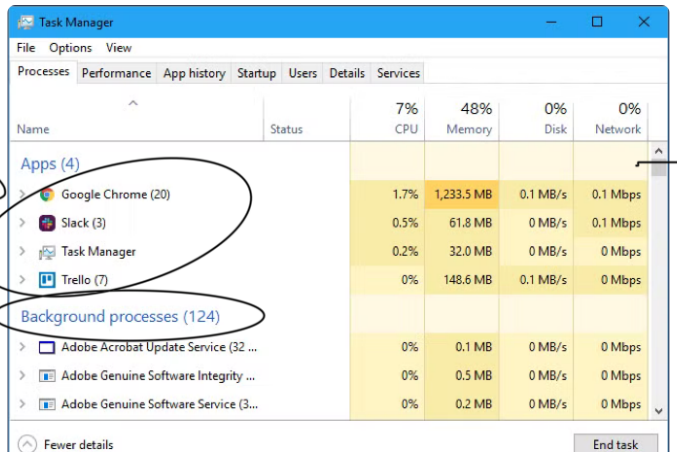
→ Boot loader
loading Operating System files
Command Line interpreter

Operating System is a very complex software that secures & manages the computer.



Operating System Layering.





Running

Currently working processes.

Explore Task Manager & Device Manager to understand the responsibility

of Operating System.

Evolution and Anatomy of Computers

Module Overview

This module introduces the **Basics of Computer Science**, serving as the foundation for more advanced topics. It covers:

- Evolution of Computers
- Components (Parts) of a Computer
- Introduction to Operating Systems
- Viewing System and Process Information

By the end of this module, you'll understand how computers and operating systems work, and how applications run within them. This understanding is crucial for any aspiring software engineer.

Evolution of Computers

The Early Days

Computers have undergone a fascinating transformation. Initially, they were massive machines, often occupying entire rooms (e.g., 10 ft by 10 ft).

- Input and output were mechanical.
- Calculations were physical and extremely specific.
- Used primarily for scientific research and calculations, such as space mission computations.

Example: Calculating the trajectory of rockets in early space missions required thousands of trials and computations.

Specialized Machines

These early computers were:

- Purpose-built (goal defined during manufacturing).
- Used only for specific tasks, much like a fan or pen cannot perform functions beyond its design.

Due to their high cost, only research institutions and well-funded labs could afford them.

Rise of General-Purpose Computers

A major breakthrough was the development of **general-purpose computers**:

- Not limited by initial manufacturing purpose.
- Could be programmed later for different tasks (e.g., accounting, gaming, communication).
- Enabled by the invention of **programming languages**.

Analogy: LEGO Blocks

General-purpose processors are like LEGO blocks:

- Fixed-design toys = purpose-specific computers.
- LEGO blocks = general-purpose computers (customizable via programming).

Microprocessors and Evolution

- **First microprocessors**: Intel 8085 and 8086, often taught in CS and Electronics courses.
- Instruction sets allowed basic operations via programming.

Processor Families

- **Intel x86**
- **ARM architecture**

Used in everything from personal computers to smartphones.

Computer Manufacturers:

- **Microsoft**: Focused on software and OS.
- **Apple**: Controlled the full stack – motherboard, processor, OS.

Modern Computing Landscape

- **Personal Computers**: Laptops and desktops using Intel, AMD, Apple (M1, M2, M3).
- **Servers**: Same architecture, but with higher core counts (up to 100+ cores).

Ubiquity of Computers

Modern computers are everywhere:

- Phones, watches, traffic systems.
- Used in edge computing, AI-powered monitoring (e.g., traffic violation detection).
- Growing presence in Industry 4.0/5.0 and IoT (Internet of Things).

Future of Computing

- Software spending expected to double (from 5% to 10% of global GDP).
- Increasing demand for software engineers.
- Penetration into sectors like education, banking, healthcare, defense, agriculture, and manufacturing.

This is still the early phase of the software revolution, and opportunities abound.

Anatomy of a Computer

To understand how computers work, we must know their parts. This helps us see where programs run and how hardware behavior affects software design.

Key Components

1. **CPU Cabinet**: Enclosure containing all major parts.
2. **Motherboard**: Main circuit board.
    - Hosts the processor and other components.
3. **Processor (CPU)**:
    - The brain of the computer.
    - Executes instructions from software.

Input/Output Devices

- **Input**: Keyboard, mouse, microphone, scanner.
- **Output**: Monitor, speakers, printer.

Storage and Memory

- **RAM (Random Access Memory)**: Temporary, fast-access memory for active tasks.
- **Storage (HDD/SSD)**: Long-term storage for files and software.

Other Components

- **Power Supply Unit (PSU)**: Converts electricity to usable power.
- **Cooling Systems**: Fans or liquid cooling to regulate temperature.

Operating System (OS)

An OS is system software that:

- Manages hardware resources.
- Provides a user interface.
- Acts as a bridge between hardware and applications.

Common Operating Systems

- Windows

- macOS
- Linux (Ubuntu, Fedora, etc.)

OS Responsibilities

- Process Management
- Memory Management
- File System Handling
- Device Management
- User Interface

Viewing System & Process Info (Basic Commands)

Windows (Command Prompt)

```
systeminfo
```

Displays detailed configuration info.

```
tasklist
```

Shows running processes.

Linux/macOS (Terminal)

```
uname -a
```

System info.

```
top
```

Live view of processes.

```
ps aux
```

Detailed process list.

Software and How It Works

What is Software?

A set of instructions (programs) that tell the computer what to do.

Types of Software

1. **System Software**: OS and utility programs.
2. **Application Software**: Word processors, browsers, games, etc.

How Software Runs

- Written in high-level languages (e.g., Python, C++, Java).
- Compiled or interpreted into machine code.
- Loaded into memory and executed by the processor.

Summary

- Computers evolved from room-sized, mechanical machines to today's compact, powerful systems.
- General-purpose processors revolutionized computing.
- Computers are now in every industry and nearly every device.
- Understanding computer anatomy helps in building better software.
- The operating system is the middle layer between hardware and apps.

Stay curious and observant — the software revolution is still unfolding, and there's much more to explore!

Know Your Operating System and Computer

Introduction

we explore the basics of the **Operating System (OS)**. While Computer Science students study OS in depth, non-IT students often don’t get the same exposure. This lecture offers a clear and simple overview, helpful for all engineering students.

If you wish to go deeper, consider watching a dedicated video on OS or refer to operating system textbooks.

What is an Operating System?

An **Operating System** is the *boss* of the computer. It controls everything that happens inside the system:

- It manages processes and hardware.
- It decides which applications run, when, and how.
- It ensures that resources (CPU, memory, input/output devices) are shared efficiently.

Common Operating Systems

- **Phones**: Android, iOS
- **Laptops/Desktops**: Windows, Linux, macOS
- **Servers**: Linux Server, Windows Server, Unix

Analogy: Traffic Police & Law Enforcer

- Like a **traffic police**, it controls traffic between different applications.
- Like a **law enforcer**, it stops misbehaving programs and handles permissions.

Structure of a Computer System

Think of a computer system in layered architecture:

1. **Hardware** – The physical machine.
2. **Device Drivers** – Translate between OS and hardware (e.g., for keyboard, mouse, storage).
3. **System Processes/Services** – Inner workings of the OS.
4. **Operating System Core** – Controls everything.
5. **Applications** – Outer layer, like browsers, word processors, or mobile apps.

This layered model shows how everything builds upon the OS.

Key Functions of the OS

1. Process Scheduling

Manages running applications and assigns CPU time.

Example: Round Robin Algorithm

A scheduling method where each process gets a fixed time slice in turn.

```
[Process A] → [Process B] → [Process C] → [Process A] → ...
```

Each switch happens so fast (in milliseconds) that it feels like all programs are running simultaneously.

2. Memory Management

- Allocates memory to programs.
- Ensures one app doesn't interfere with another.

3. Permission Handling

When an app needs access (e.g., contacts, camera, location), the OS asks you to approve it.

4. Device Management

Handles input/output devices through drivers.

5. Resource Sharing

Shares resources like CPU, memory, network, and storage across all programs.

Why is an OS Necessary?

Without an OS:

- You'd have to manually write code to manage memory, devices, and process execution.
- Running a modern computer would be nearly impossible.

Real-Life Story: Building an OS in College

The speaker recalls an attempt to build a simple OS in their third year of engineering:

- Started with understanding **boot loaders**.
- Earlier computers booted from floppy disks (1.44 MB) or CDs.
- BIOS (Basic Input/Output System) checks hardware and then looks for the boot loader.

Boot Process Flow:

1. Power ON → BIOS runs.
2. BIOS checks hardware (beep sound confirms success).
3. BIOS looks for **boot loader** (usually 512 bytes) in the boot device.
4. Boot loader loads OS into RAM.

5. A simple **command-line interface** starts.

At the time, they wrote a basic input/output system that accepted commands like `date`, `time`, or basic calculations.

Though they thought it was a full OS, they later realized real OSes are far more complex.

Types of Operating Systems

1. **Open Source Operating Systems**

- Examples: Linux, Android, Unix
- Source code is available online (e.g., GitHub).
- Usually written in **C or C++**.

2. **Proprietary Operating Systems**

- Examples: Microsoft Windows, macOS, iOS
- Owned by companies like Microsoft or Apple.
- Source code is not publicly available.

Summary

- The Operating System is essential to every computer and mobile device.
- It handles memory, permissions, device access, and process scheduling.
- Without it, using a computer would require writing low-level code.
- OSes like Linux are open-source and customizable, while others like Windows or macOS are closed-source.
- Even a simple boot loader is a fundamental step toward understanding OS internals.

Understanding the OS helps you become a better programmer, system designer, or tech-savvy engineer.
