Basics of Computer Science

Day 2
12/07/2025

- Evolution of software Industry

The Software word Came in picture roughly around 1950 By John.

mechanically designed

Machine program

Thought of Hardware & Software Came into picture.

First Programming Language

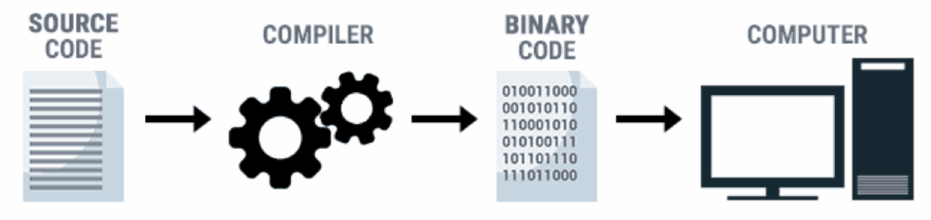
Binary language

1
0] Binary language for Computers

"1010 10" → Binary language

Any type of Data / Content in the Computer is processed in the form of Binary (0 or 1) or stored in the form of Binary.

Ex :- Text, Image, Number, Audio ... Ext



Initially when the Binary language is used for programming But the Binary language is very difficult to write Programs So the Assembly language is used for programming.

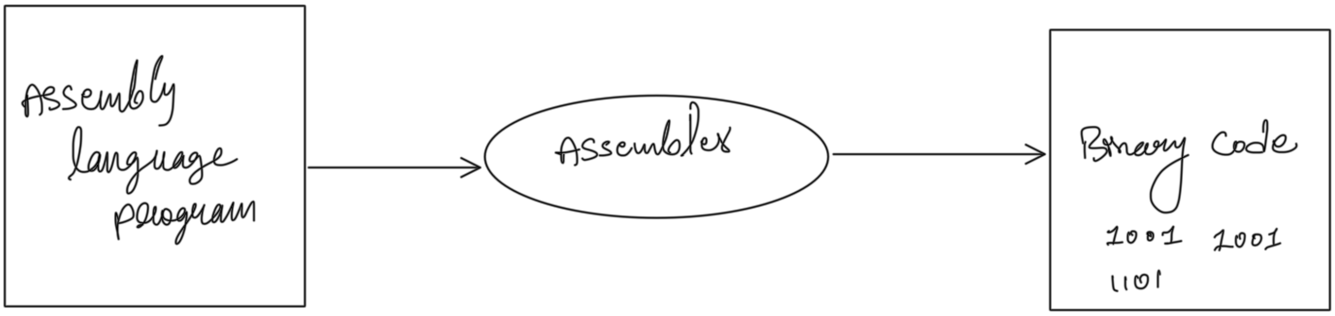
The Assembly language is Easier to write compared to writting program in Binary.

Binary language

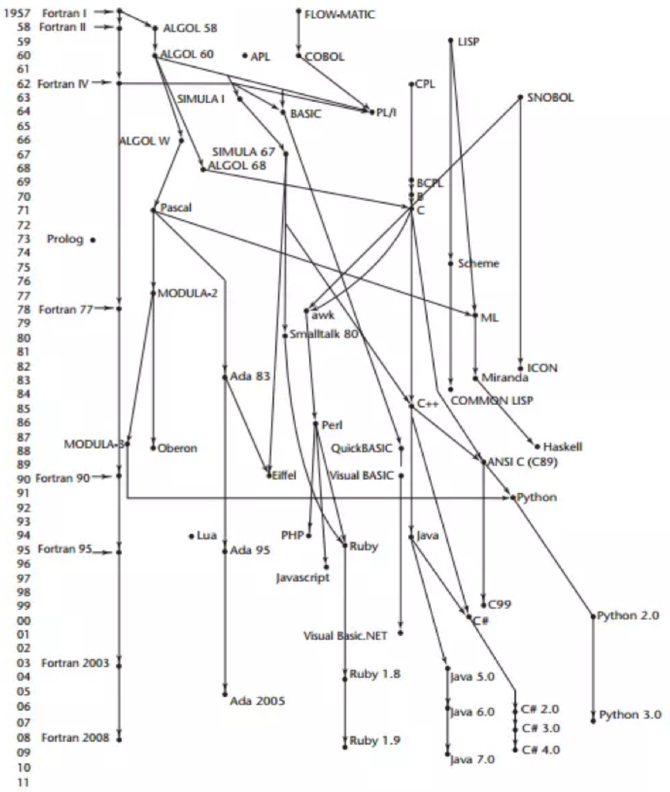
⇩

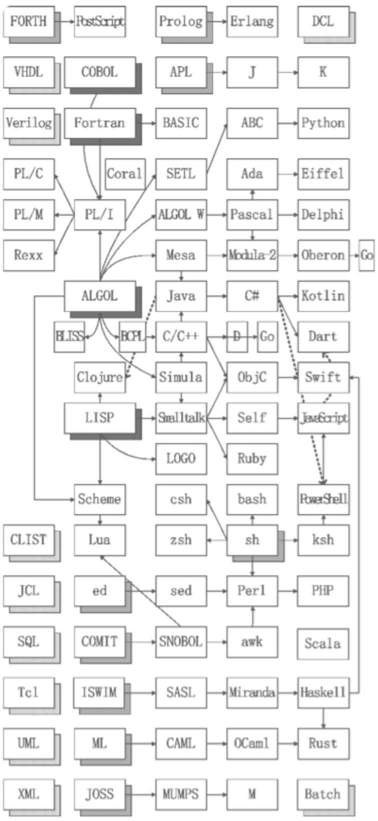
Assembly language

```
MOV  AX 01
SUM  AX
SET
JMP
```



In 1960 or 1970 programming languages like Fortran prolog Basic programming languages came into picture



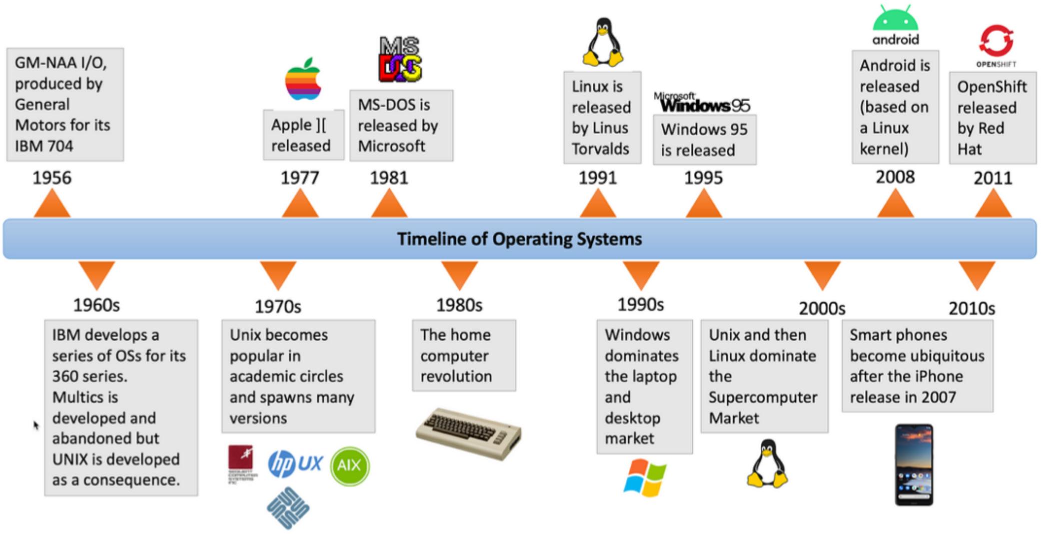


After the arrival of new programming languages in the programming

Demand on Operating System Increases.

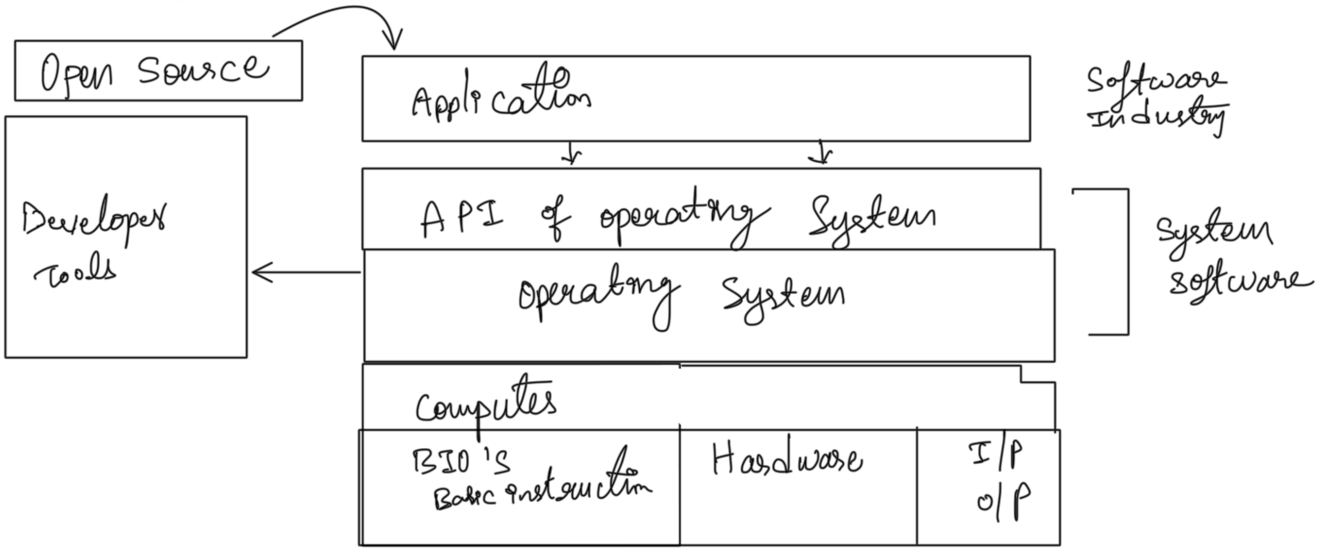
Initially the software Industry was aiming for improving languages & their usage By the programmers and aimed to increase the No of programmers in the large ratio.

Evolution of Operating System.


GM-NAA I/O, produced by General Motors for its IBM 704
1956
Apple][released
1977
MS-DOS is released by Microsoft
1981
Linux is released by Linus Torvalds
1991
Windows 95 is released
1995
android
Android is released (based on a Linux kernel)
2008
OpenShift released by Red Hat
2011
Timeline of Operating Systems
1960s
IBM develops a series of OSs for its 360 series. Multics is developed and abandoned but UNIX is developed as a consequence.
1970s
Unix becomes popular in academic circles and spawns many versions
1980s
The home computer revolution
1990s
Windows dominates the laptop and desktop market
2000s
Unix and then Linux dominate the Supercomputer Market
2010s
Smart phones become ubiquitous after the iPhone release in 2007

Operating System Becomes middle man. for Handling API's and other Operations.

Operating System makes all the work

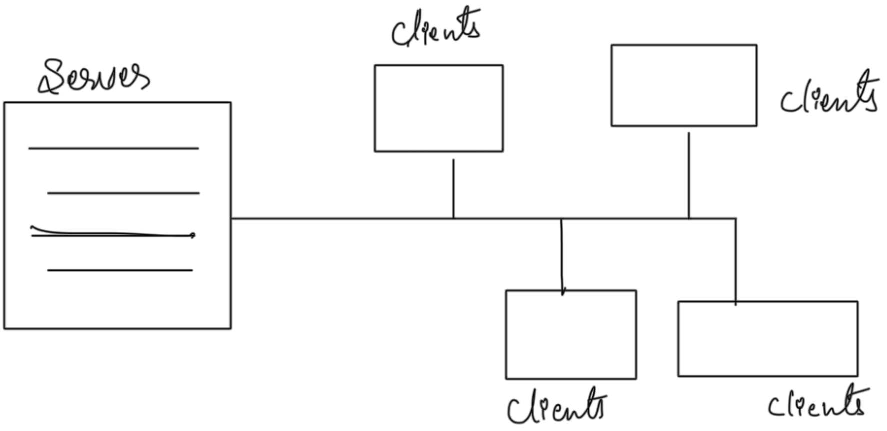


The Industry which develope Software Applications, the applications which can Run By Operating Systems using its API's

This Application development Industry called as Software Industry.

Developer Tools helped to Build Application in faster process. The process of Developing software Applications Becomes faster

Network / Internet Introduced client Server pattern

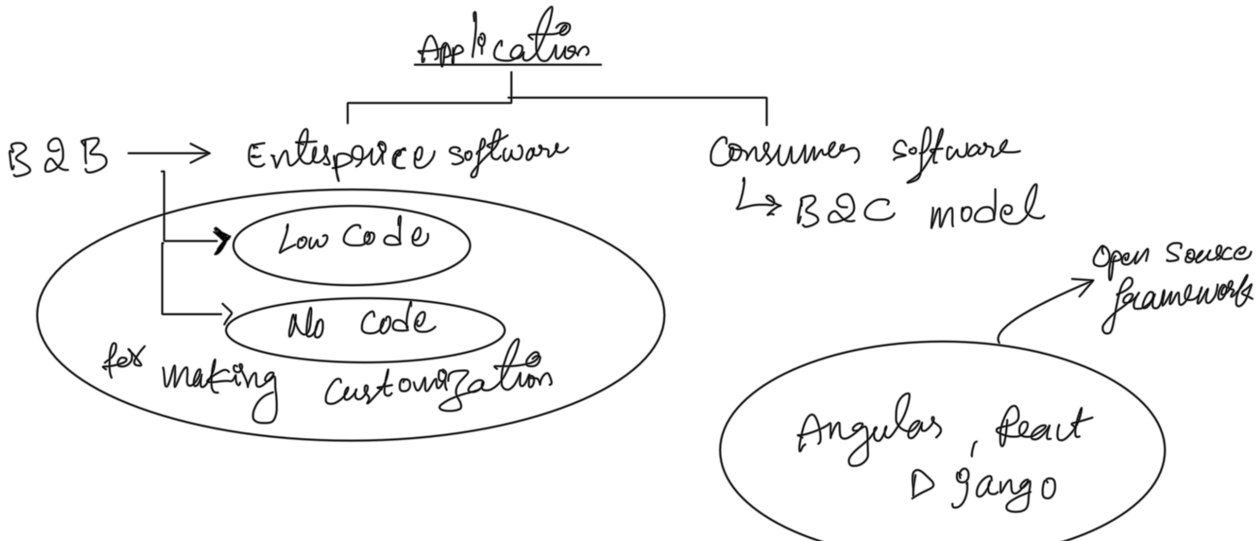


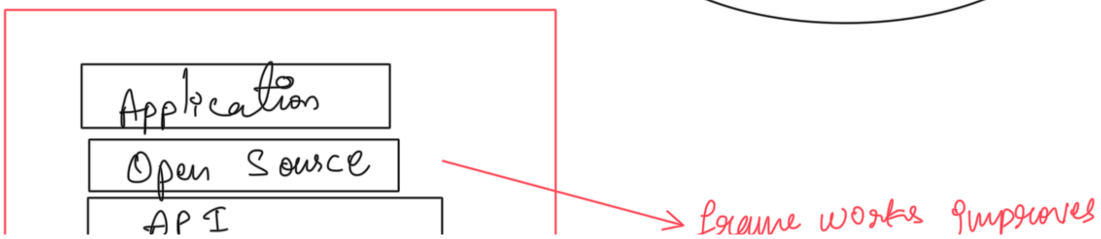
Various permutation & combinations on Computer Network were implemented.

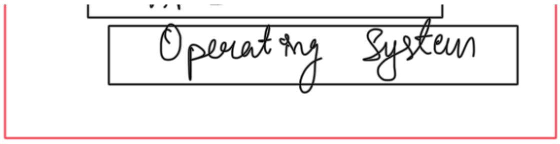
New "Open - Source" trend is evolved

Open Source improved Application development, enchancement of the Application Basically contributions are increased on a Application development.

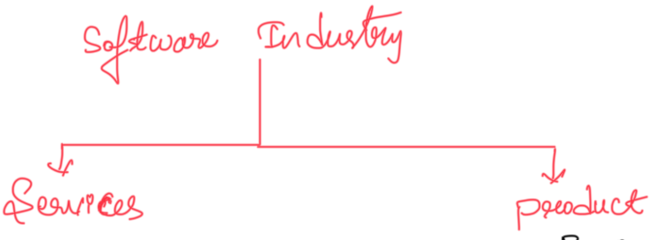
No code - Low code trend Evolved







reusability of code



IT services for others Businesses

IT Services are B2B

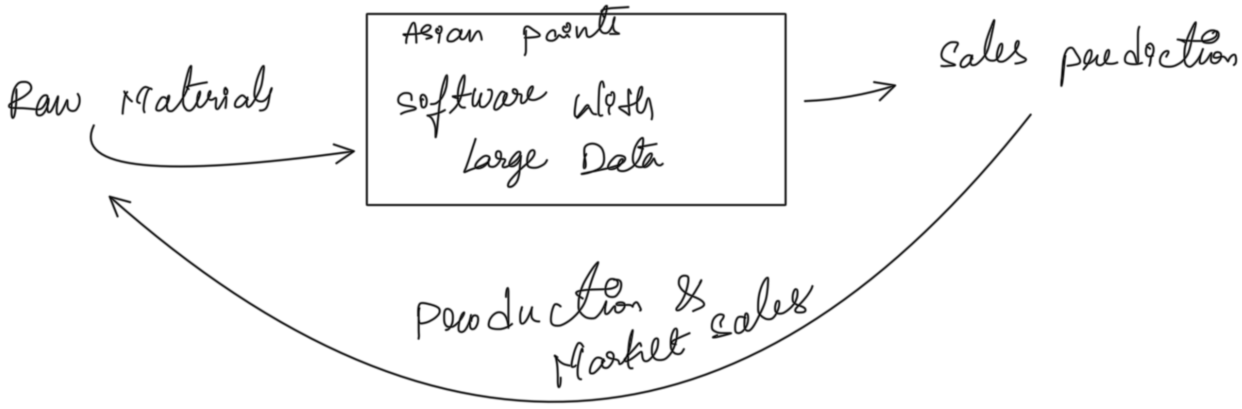
Business to customes (B2C)

UPI
Delhivery
shopping
Travel

Customes is deriving this service Based Industry

Meet any Industry using of software is must to grow in the competative market.

First Super Computer in India was purchased By "Asian paints" in 1950



World's GDP 5% is spent on IT Industry

In next 10 years it will be increased to 10%

Total revenue of software 1 Trillion $

Total Revenue Stretching 1 Trillion $

Top 10 Companies in the world are majorly from Software Industry.

Innovation feeds Innovation which increases Human productivity

Human productivity saves time

Evolution of Software Industry and Roles – Part 1

Introduction

we'll explore the fascinating journey of the software industry: from its origin to where it stands today. We'll also discuss different roles, types of software, and how industry dynamics have shaped its growth.

Understanding this big picture is essential — it's like having a GPS for your tech career.

How Software Started

- The term **"software"** was coined around **1950** by a mathematician named **John**.
- In early computers, programs were tightly coupled with the machines — mostly mechanical.
- Around the 1950s, the idea emerged that software could be separated from hardware.

The Binary Language

- Computers operate using **binary (1s and 0s)**.
- Whether it's a **photo, video, text, or number**, it's stored and processed as binary.

Example

```
# In Python, converting decimal 10 to binary
bin(10)  # Output: '0b1010'
```

Evolution of Programming Languages

Assembly Language

- Replaced binary programming.
- Used simple instruction sets like:

  ```
  MOV AX, SUM
  ADD AX, BX
  ```

- Required an **assembler** to convert to machine code.

High-Level Languages (1960s–70s)

- **Fortran**, **COBOL**, and **BASIC** were developed.
- Languages became more readable and easier to use.
- This made programming accessible to more people.

Rise of Operating Systems

Why Operating Systems Were Needed

- Originally, programmers had to manage all device interactions (input/output, printing, etc.).
- Example: 5000+ lines of code were needed to print a file.
- With OS and APIs, the same task takes only 4–5 lines.

Architecture

```
[ Applications ]
     ↓
[ Operating System (APIs) ]
     ↓
[ BIOS + Hardware Drivers ]
     ↓
[ Computer Hardware ]
```

- **BIOS** has basic instructions.
- **OS** bridges between hardware and software.
- **APIs** handle file I/O, networking, rendering, etc.

System Software vs. Application Software

System Software

- Operating systems like **Windows**, **Linux**, **Unix**, **macOS**.
- Developed by Microsoft, Apple, IBM, and open-source communities.

Application Software

- Runs on top of OS using provided APIs.
- Two types:
  1. **Enterprise Software** – Salesforce, SAP, Walmart systems (B2B).
  2. **Consumer Software** – Ola, Uber, Swiggy, Amazon (B2C).

Developer Tools and Open Source

- **Microsoft** and others created OS + Developer Tools.
- Tools made it easier to build applications.
- **Open-source** libraries accelerated development.

Example Earlier, database operations required 30+ lines of code. Now:

```
import sqlite3
conn = sqlite3.connect('data.db')
```

This simplification was made possible by open-source libraries and reusable client code.

Industry Trends: No-Code & Low-Code

- **Low-code**: Minimal coding needed to customize software.
- **No-code**: Drag-and-drop interface, no programming required.
- Common in enterprise workflows where custom solutions are expensive and time-consuming.

Software Architectures

Standalone Software

- Installed and run locally.
- Doesn't communicate with other systems.

Client-Server Model (1980s–1990s)

- Software runs on centralized servers.
- Clients (user devices) connect over a network.
- Enabled by the rise of the internet.

Peer-to-Peer and Distributed Systems

- Software and resources are decentralized.
- More scalable and resilient.

Software Development Models

1. Product-Based Companies

- Build and sell their own software products.

- Example: Google, Microsoft, Facebook, Apple, etc.

2. **Service-Based Companies**

- Build custom software for other businesses.
- Example: Infosys, TCS, Wipro, HCL

Services Model

- Client provides requirements.
- IT service company builds, deploys, and maintains the software.
- Most code is proprietary and client-specific.

Software in Business

Software is now essential across industries: - **Banking**: In-house teams with 5000+ software engineers. - **Manufacturing**: Asian Paints uses software for supply chain and distribution. - **Automotive**: Tesla is as much a software company as a car company.

Global Impact of Software

- In 2023, the global IT industry reached nearly **$1 trillion**.
- Tech companies dominate the list of top 10 companies globally.
- Software increases **human productivity**:
    - Automates low-skill jobs.
    - Frees up time for creative and strategic work.

The Future of Software

- We're still in the **early stages** of this revolution.
- **AI, ML, robotics, automation** will further increase productivity.
- **Government and public sector** are embracing IT for efficiency.

Example

- Bangalore uses AI to detect traffic violations in real-time.

Summary

- Software evolved from binary and mechanical systems to highly abstracted, API-driven models.
- Operating systems and developer tools reduced development time.
- Open-source tools and no-code/low-code platforms accelerate innovation.
- Software industry spans both consumer and enterprise domains.
- Product vs. Service-based models define company structures.
- Global adoption of IT continues to increase human productivity and transform industries.