Hardware and Software❤️‍🔥

History of computing

2400BC - First computer invention
1830 Charles Babbage - analytical engine ,difference engine(to calculate tables of polynomial values)
1937 - first computer without wheels and cogs
1975 - first personal computer

Babbage designed (but never built) the first general-purpose programmable computer - the analytical engine

Early Chinese abacus was used to add, subtract, multiply and divide

Mechanical calculators - Schickard, Pascal, Leibniz etc

1800s - Jacquard used punched cards to control a loom
Hollerith used punchcards for the 1890 US census(his company became IBM)

Vannever Bush - built a differential analyser that used wheels/discs to perform integration(took the analytical engine that Charles Babbage designed and built the analyser)

1946 - ENIAC (one of the early programmable digital computers, using vacuum tubes for computation and patch cables for manual programming
1943/1944 - 2 university of Pensylvania professors John Mauchly and J.Presper built the ENIAC which contained the vacuum cubes

1937(Atanasoff-berry) - (professor of physics and math attempted to build the first computer without the cams ,belts etc)

ENIAC - Electronic Numerical Integrator And Calculator

1960s - first transistors, integrated circuits, microprocessors where created .This revolutionised computing making smaller, cheaper and more general-p

- As the sizes got smaller, you could store more information on those devices
- Generally the first computers in the 1950s where not easy to use only the Ancient geniuses knew how to use them
-

After 1975 ,after the first computer coming up, few companies started

designing smaller computers making computers usable to just about anyone

From an age where you had to be a genius to operate a computer to an age where you could use your own

Today we have more powerful devices ,more powerful than the ENIAC

Computers have changed in the last 50/60 years. As we are able to store and process more information using smaller, faster devices as time went on.

Hardware

Hardware refers - to the physical parts of the computer(often referred to as computer components and peripherals)

E.g Motherboard, Hard Disk/Drive

Software refers - to the set of instructions given to a computer to execute one or more tasks(sometimes referred to as programs)

E.g. Microsoft Office , Firefox

The Von Neumann Architecture
This describes how a conceptual computing device works:

- Memory stores data and instructions
- Control Unit (CU) obtains and executes instructions
- Arithmetic Logic Unit (ALU) does calculations
- Accumulator is internal ALU storage for some data
- Input is process of getting data into machine
- Output is process of obtaining data from machine

CPU : microchip that performs computation. It usually contains the ALU and CU.

Memory (primary storage): microchips that store data which can be accessed while computer is switched on.

Random Access Memory - volatile and modifiable(most important memory in a computer).It is directly connected to the ALU and CU
Read- only Memory(ROM) - cannot be changed(used to check if computer is working properly)

Hard drive, Floppy drive (secondary storage): store data on magnetic discs permanently i.e., the data is not lost when the computer is switched off.

▪ **Input/Output devices:** transfers data from operator to machine and vice versa.

▪ **Operating System:** software system that manages resources on computer and executes application programs, e.g., Windows XP, Ubuntu Linux.(basically controlling your computer system)

Software

Muhammad ibn Musa al-Khwarizmi (Persian mathematician)
Early work on Algebra
Later work on algorithms

The father of Computer Science - Alan Turing

Defined what can be called an algorithm and showed the universality of algorithms.
▪ Church-Turing thesis
▪ **Defined a Universal Computer**
-Turing Machine
▪ Defined **how we test for AI**
-Turing Test
▪ Part of WWII code-breaking team that deciphered Enigma messages.
▪ Harassed by UK government because he was gay ▪ **committed suicide in 1954.**

Grace Hopper

Mathematician, computer scientist, Admiral in US Navy

Major contributions:
▪ High-level programming language
-COBOL
▪ Invented first compiler
▪ Conference and awards in her honour
▪ **Popularized the term "bug"**
▪ All this and she started coding at age 37...

Women in Computer Science

Ada Lovelace (1815 - 1852)
- First computer program on Babbage's analytical engine

- Margaret Hamilton (1936 -)
  -Lead s/w designer for Apollo missions

- Adele Goldberg (1945 -)
  -Part of the Smalltalk team at PARC

- Anita Borg (1949 - 2003)
  -Advocacy for technical women

What does the CPU understands

Machine Code is the only language a CPU understands
Each interaction is a sequence of numbers

On x56 CPUs the instructions have variable lengths

The Operation System

Manages resources on computer.

- Executes on startup (boot):

We cant change ROM

-BIOS ROM has instructions to load OS - fixed in hardware!
- Disks are checked in order defined by hardware.
-If OS machine code is on a disk, load it into memory and start execution.
-OS takes over and allows users to select and run their programs until computer is shut down.

Instructions are written by human and are converted into machine code so your computer can understand

Machine Code is a low level language.

- ONLY language CPU can understand.
- Different MC for every CPU!

- **Low level languages** are easier for a machine to understand, and often difficult for a human.

- **Assembly language expresses machine code symbolically - so humans can write programs more easily. (Allowed humans to control the computer in a way that was easier for humans to understand)**

- Example (quit a program):
  - decimal: 108 76 205 33
  - hexadecimal: B4 4C CD 21
  - assembler: MOV AH,4Ch INT 21h

Programming in Assembler

- Write program in text editor.
- Save program in file.
- **Assemble** source code into object/machine code. ▪ tasm hello.asm
- Optionally link multiple files together and create executable understood by OS.
  tlink hello.obj
- Execute application in OS. ▪ hello.exe

Pros/Cons of Assembler

Pros

- Matched machine code so can do whatever CPU can do.
- Very fast execution of programs.
- Can be used on obscure CPUs.

- **Cons**
- Difficult to program.
- Programs are very long.
- Programming is slow process and prone to errors

High Level Language(python, java ,C++)

High level languages are easier for humans to understand.

- We need to convert programs in high level languages to low level languages so computers can understand.

- 2 common approaches:
- Compile
-Interpret(reads each statement/slower)

Python has an interpret and Java has a compiler

Interpreter

Interpreter reads each statement and executes an equivalent set of machine code instructions.

Compared to compiled programs:

- Easier to program - no compile step.
- Programs run slower - source code must be processed every time.

Types of High Level Languages

- Procedural/Imperative, Object-oriented - programs are specified as exact sets of instructions to execute.
  -Python, Java, C++

- Declarative/Logic - programs are rules and facts that are processed by an engine.
  -Prolog, XSLT
- Functional - programs are collections of functions that are applied and composed to solve problems.
  -Haskell, Lisp