

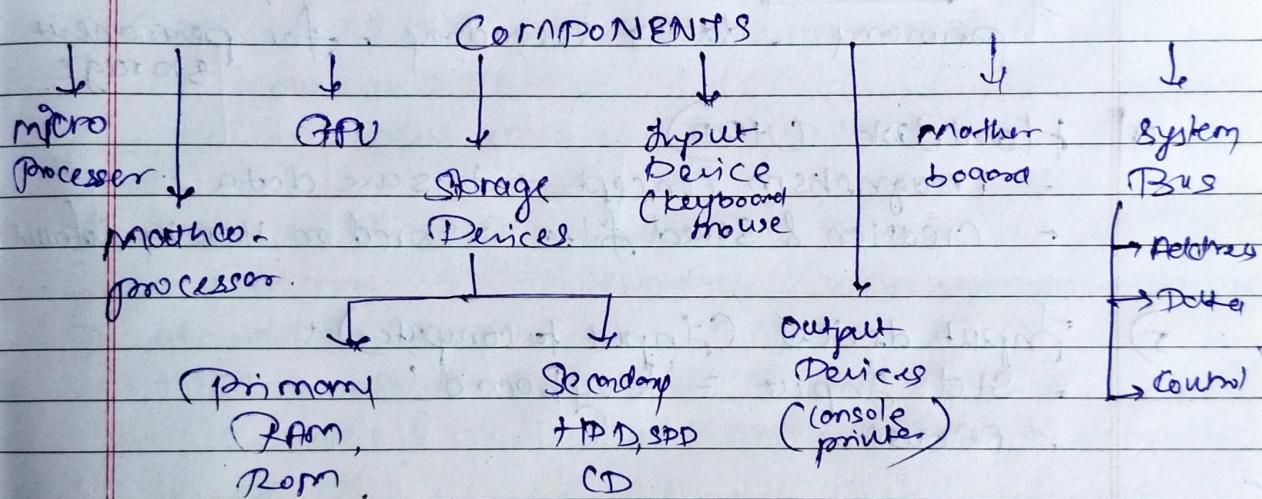
Assignment 1

1) What is computer?

- Computer is an electronic device.
- It is used to perform simple as well as complex operations and problems with help of input & output devices & its CPU.

2) What are different components of computer?

- Computer is formed by different components as follows:



1) microprocessor (MP):

- Known as - Central Processing Unit
- Instruction Decision making, execute
- Understands Binary language (0, 1)

2) Mathcoprocessor:

- Sub unit of MP
- Executes Arithmetical & mathematical instructions.

3) GPU (Graphical Processing Unit)

- Dedicated processor for graphical processing
- Instructions related to display of graphical data.

4) Storage device

Store all types of data.

A) Primary

Accessible to MP directly

i) RAM (Random Access memory)

- Primary electrical storage device

ii) ROM (Read only memory)

- Every ROM before is RAM.

- Permanent data in RAM = ROM

B) Secondary

primary \rightarrow secondary - for permanent storage

T) Hard disk (HDD)

- Magnetism concept \rightarrow to save data.

- Creation & saved files \rightarrow stored on HDD by default

5) Input devices (Input to computer):

- Std. Input = 'Keyboard'
- Mouse

F) Output devices:

- Std output = Console

7) Motherboard:

- Printed circuit Board (PCB)
- connects all components

8) System Bus:

- Bunch of wires used to connect devices to motherboard.

- Types : 1) Address Bus

- 2) Data Bus

- 3) Control Bus.

Q) What is Storage device and types of storage device.
Storage Devices :

- These devices store all types of data.

A) Primary :

- Accessible to microprocessor directly.

1) RAM (Random Access memory) :

- Primary electrical storage device.
- Program transferred in RAM for execution.

2) ROM (Read only memory) :

- Before it is RAM.
- Permanent data in RAM \Rightarrow ROM

3) Secondary Devices :

- Primary \rightarrow Secondary \rightarrow for permanent storage

D) Harddisk (HDD)

- Magnetism used.
- created & saved file - stored on HDD by default

E) SSD (Solid State Drive)

- Electricity required
- More expensive than HDD, as life span is more
- \Rightarrow SSD + Harddisk = Fusion Drive

Q) What is meant by Toolchain or x86 Toolchain?

\rightarrow o Toolchain = Set of software (tools) used to convert human understandable into machine understandable program (binary)

o x - models of particular Intel family

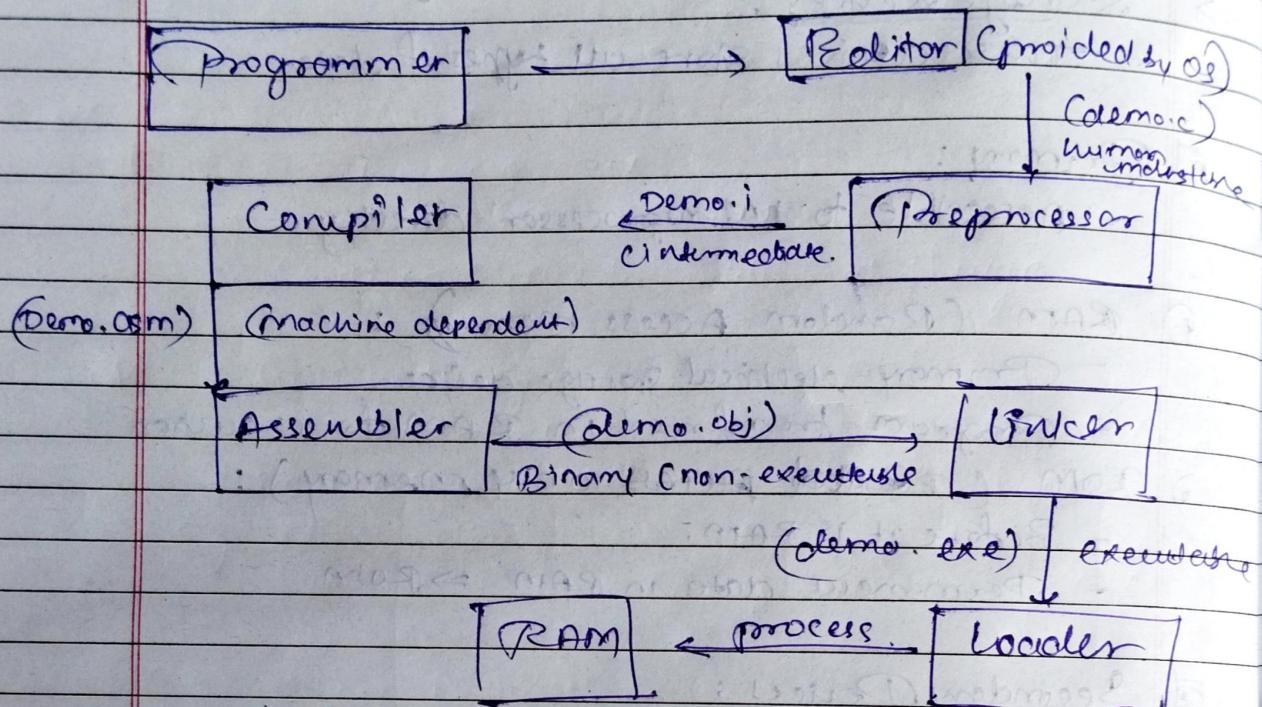
e.g. 80186, 80286 . . . 80x86

• x86 Toolchain

- tools connected in a 'chained format'

- Final output - produced by linker.

→ ToolChain



X - 86 Toolchain

- Every Programming language has its own toolchain, Connected in 'Chain Format'.
- Output of one tool is input for next one.
- When executable file (.exe) is stored in HDD, Loader loads it to RAM so CPU can access it.

Q1) What is CPU register? What are types of it? and explain use of each CPU

- 1) CPU Register : Type of computer memory built directly into processor or CPU, used to store & manipulate data during execution of instructions.
- 2) Size : 8 bit, 16 bit, 32 bit, 64 bit

types

- 3) AH, AL - Arithmetic / Accumulator register.
(logical operations & calculations)
- 4) BH, BL - Base register (provides an index into connected memory-mapped resource)
- 5) CH, CL - Count Register
- 6) DH, DL - Data Register (store data fetched from memory / input-output operations)
- 7) SP - Stack Pointer
Points to top of stack, which is a region of memory used for temporary storage during function calls & other operations.
- 8) BP - Based Pointer
used for based indexed or register indirect addressing
- 9) Destination index - DI (used to store offset address of destination)
- 10) SI - Source Index
used to point address of data & as a source in string-related operations.

6) What is operating system & its different tasks.

Operating System (OS)

- Already present in RAM, before any program runs
- It is a system software managing computer hardware & software resources & provides common services for computer programs.
- Installed in HDD
Transferred to RAM
Booted on
Boot

7) Tasks

File Management

- Adding, deleting, managing the files

5) Process management

- process has - 'lifespan'
- program in run-time mode is managed

6) Memory management

e.g. Hotel & Restaurant booking system

- Creating space for additional storage

7) CPU Scheduling :

- Priority given to process which is urgent (importance)
 - makes system efficient, fast & fair
- Eg. M.P (Doctor) → CPU scheduling → Priority → time (Receptionist) (patient)

8) Hardware Abstraction :

- Acts as a interface between hardware & OS.
- Unnecessary things are eliminated.

Explain working of each tool from ToolChain i.e.

Editor, Preprocessor, Compiler, Assembler, Linker, Loader



1) Editor -

- Provided by OS
- Program is written in the editor.
- After writing program file is saved in HDD with name → demo.c (Human understandable & Readable)

2) Preprocessor -

Creates input demo.c & generates expanded version of file.c

File created - demo.i (i = intermediate code)

Eg. $\#include \text{ "stdio.h" }$

3) Compiler -

Input - demo.i gets converted into demo.asm
(Assembly language) → machine dependent

4) Assembler -

Input - demo.asm gets converted from machine dependent to → demo.obj (machine understandable)
(non-executable).

5) Linker - (प्रोजेक्ट)

Input - demo.obj → converts to executable file (demo.exe)

6) Loader

Transfers file from secondary storage (HDD) to primary storage (RAM), to be executed by MP.

87

→ Above is the diagram of 'Motherboard'.

Step 1: Programme writes program by using editor (.c)

2: Compiler - Internally contains (preprocessor, compiler, assembler)

a. obj divided into 3 parts:

1) Text - contains compiled instructions of a program in binary format

2) Data - memory for global variable used in program

3) Symbol Table containing info about symbols (special characters) used in program.

Step 3: Linker

- links files with it's dependencies
- creates a .exe file (executable)

Step 4: Stored inside HDD → Loaded in RAM. [Booting]
↓
Loader of OS is responsible.

- Step 5:
- Inside .exe file the PH (Primary Header) is added.
 - Contains info about a .exe file.

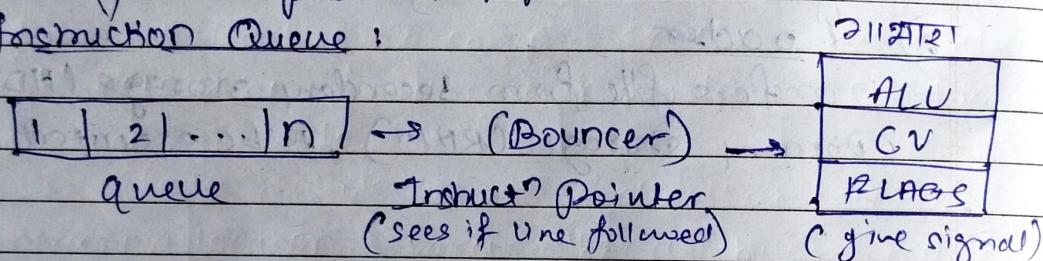
Step 6: Process (Hello process) - Loader loads file in RAM.

```

    graph LR
      Process --> Text
      Process --> Data
      Process --> Stack[Stack (Stack frame)]
  
```

Step 7: RAM is not responsible to execute the process
 \therefore Process passes to MP.

Instruction Queue:



Step 8: Inside MP (CPU) there is no sufficient memory like RAM due to which direct loading of text, data & stack section is not particularly possible.

- Section divided into \rightarrow 'Segments'
- 1) Text \rightarrow Code Segment (CS)
- 2) Data \rightarrow Data Segment (DS)
- 3) Stack \rightarrow Stack Segment (SS)

Note: If all above segments are full then the 'RS' is used RS (CS are further used).

Step 9: Data from segments transferred \rightarrow from instruction queue

- Single instruction fetched at a time & forwarded that info execution purpose.
- IP responsible to fetch that single info.

Step 10: Instruction forwarded to ALU (Arithmetic Logical Unit)
 \therefore instruction is related to Arithmetic operation otherwise.

forwarded to MP (Cloud Unit),
PLAQS Register.

Responsible to indicate internal status of CPU.

Step 11: Info stored in CPU register.

These are multiple CPU register, depending upon requirements.

1) AH, AL - Arithmetic / Accumulator.

2) BH, BL - Base Register

3) CH, CL - Count - 1

4) DH, DL - Data - 1

5) SP - Stack Pointer

6) BP - Base Pointer

7) SI - Source Index.

8) DI - Destination Index.

* Inside all CPU register, execution of instructions gets performed.

Step 12: Output of instruction is forwarded to OS with help of IDB (Internal Data Bus)

After getting output OS will display it on Console.

107

i) Text Section - Contains compiled machine code (instructions) of program. Typically read-only & shared among processes.

E.g. 'int add(int a, int b) { return a+b; }'. code generated for this funcn stored in text section.

a) Data Section - Stores global & static variable that have been initiated by programmer.

b) initialised Data Segment

Uninitialised Data Segment

e.g. 'int x=5;', 'x' would reside in initialized data segment. If you declare 'static int y;', 'y' would be placed in BSS segment.

Q) Stack Section:

- Manages function calls, local variables & return addresses. It grows & shrinks with functions
- For each funcn there is - 'Stack frame'

Q) What are contents of Physical header?

- 1) Physical Header is a complex structure that contains all necessary information for windows OS to load, manage & execute the file.
- 2) It includes details about architecture, sections, memory allocation & various other attributes critical for execution.
- 3) Also it contains information about .exe file.
e.g. Pack of bread contains information about the quantity of carbs, fat, protein, etc.