Computer Architecture

Assignment : 1

Q what are the components of computer ?

→

Computer is made up of several components. There are classified as two types
Hardware components
Software components.

*Hardware components are
1) central processing unit
2) Primary Memory
3) Secondary Memory
4) Motherboard
5) Input devices
6) Output devices
7) Graphics processing unit

*Software components are
1) Operating system [OS]
2) Application software
3) Programming language and libraries.

1) central processing unit is brain of computer responsible for executing instructions and performing calculations.

2) Primary Memory
- RAM [Volatile Memory]
- ROM

RAM [Random Access Memory]
ROM [Read Only Memory].

3) secondary Memory
[Storage Memory]
[Non-volatile Memory].
- HDD - hard disk drives
- SDD - solid state drives

4) Motherboard
- Main circuit board
- provides connecting and communication between different parts of the computer.

5) Input devices
Mouse, keyboard etc

6) Output devices
Monitor, printer

7) GPU → rendering graphics useful for games, video editing.

*Operating system [OS]
- core software that manages the computer's hardware and provide a user interface
  - Windows
  - MacOS
  - Linux

Q What are the types of storage devices

→

HDD = Hard disk drive
SDD = Solid state drive
USD = Flash drive
drives

There are major three types of storage devices that are HDD, SDD, Flash drives

HDD = slower than SDD. Higher cost than flash drive. longer in size

SDD = full of SDD is solid state drive. faster than HDD, mostly, now a days used in laptops..

* SDDs are commonly used as primary storage in Modern computers.

USD drive. external storage devices. Flash drive, pendrive small storage. Medium, Avg speed. They are connected to via USB ports and are often used for transferring files between devices.

Q What is meant by x86 Toolchain

- x86 Toolchain means The Toolchain that includes compiler, assembler linker, loader, editor Toolchain components together to compile, Assemble, link, loader and execute the code written in various language including C, C++, Java python and many more. So the x86 Toolchain means this above Toolchain is same for all the x86 series processor that includes 8086, 80286 80386.

The Toolchain works some in all processor of x86 series. that converts Human readable code to machine executable code.

Q what are types of CPU registers? explain use of each CPU register

→

CPU registers plays important role as instruction are stored and term in it. There are multiple CPU registers, and are used according to the require. The instructions, data are store in CPU registers in. form of segments.

There are two types of registers in CPU

RCU and ECU which are further various registor in it.

Here RCU stands for Register control unit and

ECU stands for execution control unit

Here RCU CPU register has

AH	AL	SP
BH	BL	BP
CH	CL	SI
DH	DL	DI

AH stands for

AH = Arithmatic OH
AL = Allnmation register

BH, BL are Base register
CH, CL are count register
DH, DL are data register and

SP ⇒ stack pointer
BP ⇒ Base pointer
SI ⇒ source Index
DI ⇒ Destination index

Now let discuss about ECU CPU register

ECU stands for Execution control unit.

ECU has

CS	If first three
DS	register are full
SS	the information is
ES	transferred to ES
FS	followed by FS
GS	GS.

There NO full form for this registers.

CS = Text segment get into CS [code segment]

DS = data get copied into DS [Data segment]

SS = stack get copied into SS [stack segment].

Q Explain working of each component from x86 Tool chain i.e editor, Preprocessor, compiler, Assembler, linker loader.

→

~~* step 1 that is editor~~

In x86 Toolchain there six components that work together to convert the Human Readable code into Machine executable code. from High level Machine understandable code.

The working of x86 Tool chain is same of for x86 Processor series..

The working of Toolchain starts with editor.

step 1 [Editor].

- programmer uses editor to write the program.
- The code is written in Any language, it can be C, C++, Java, python.
- Suppose code is written in C language and file is created called demo.c Here .c is the extension of the C programming language.
- The contains of the demo.c are Human Readable format. and understandable.

*** step 2 [Preprocessor]**

- The preprocessor takes demo.c as input from the editor output.
- The contains of the file are Human Readable and Human understandable.
- The file demo.c is converted into demo.i which is expanded version of demo.c
- Here i stands for intermediate code.
- size of the file is increased as compare to demo.c

*** step 3 [compiler]**

= The output of the preprocessor demo.i is input for the compiler

- The compiler converts the demo.i into the demo.ASM or demo.s
- Here .ASM and .s are extended version of Assembly language

- The compiler converts High level language code in low level language
- The converted code is machine dependent format not in Human understandable format.
- The Machine dependent format is Assembly language
- Now the file is demo.ASM as output of the compiler.

* **Step 3 [Assembler]**

- The output of the Assembler that is demo.i is input of the Assembler.
- Assembler is a software that converts demo.i file into demo.obj.
- The .obj file code is in binary format but not directly executed.
- The Assembler converts Machine dependent format into Machine understandable format that is binary format.
- The Newly created from Assembler is demo.obj.
- .obj is a extension

* **Step 4 [linker]**

= The linker links all the Necessary file to execute the code which is in .obj file and generates the obj output in format of .exe

= The .exe file is executable file.

- Here the Newly created file is demo.exe.

* **Step 5 [loader].**

= The loader loads the demo.exe file from secondary memory into Primary Memory that is RAM

- The loader is responsible to load the .exe file from Harddisk to RAM.
- After the file is loaded it is considered as the process. and it is executable file and get executed with the help of OS.

Q what are the task of operating systems.

= operating system is considered as a software which performs five tasks

- file Management
- process Management
- Memory Management
- CPU scheduling
- Hardware Abstraction

- All this five task are exactly same in any type of operating system. Windows, linux, OS [MAC].
- When we start our laptop our desktop the OS get loaded from Harddisk into ram.

Q what are the content of the primary header?

→

The primary header in executable file has the information about the executable file, it maybe address ~~of~~ to where file must be located.

for example if the file is image format like JPEG or PNG, then the primary header might contain information about the image dimension, colour space, compression method and many more.

Q what is mean by Text, data, stack section. ?

→

Here

TEXT = Text includes All the compiled instructions of a program in the binary format

DATA = The section contains the Memory for global variables used in the program.

Stack = Stack section contains all the info. about the function written in our program.

Q Explain each step of below diagram? [computer Architecture].

=> ~~computer architecture of~~

The below diagram contains several steps to execute a file. consider the file demo.c to be excuted.

Now

Step 1

- we store the file with the extension demo.c [dot c] bcoz program is written in c language
- consider file Hello.c code is written with the Help of the editor and saved

Step 2

- The file is passed to the compiler.
- In the diagram the Name compiler internally contains preprocessor, compiler, Assembler.
- The output of compiler is Hello.obj
- The obj file is divided into three parts Text, Data, Symbol Table

Step 3

- Now we pass the Hello.obj file towards the linker.
- linker links all the Necessary file required to Hello.obj file to execute.
- The output of linker is Hello.exe which is executable file
- The linker converts the Hello.obj to Hello.exe and Machine dependent format into the Machine understandable format that is exe. binary format
- The linker is divided into ~~three~~ four parts
  - primary Header
  - Text
  - Data
  - Symbol Table.
- **primary Header** = contains the information about the file.
- **Text** = It contains the compiled instructions of a program in the binary format.
- **Data** = This section contains the Memory for global variable used in the program.

- Symbol Table = It is table which contains the information about the symbols. [variable] which are used in our program.

Step 4

- The executable file is Hello.exe which is storage in Harddisk and with the help of loader the file is loaded in RAM, primary Memory.
- loader is responsible to the loader the file in order to execute the file Hello.exe

Step 5

- The file is loaded in to RAM with the help of loader Now the file Hello.exe is known as process [called].
- The file is process Now and divide into three parts segments TEXT, DATA and stack

Stack = It contains all the information about the functions used in our program.

Step 6

= The file, information data are divided into segments and pass to CPU and data, instruction are stored in Registers.

- The TEXT segments are loaded to CS in ECU in CPU Register Here the CS is code segment and ECU Execution Code unit
- The DATA segments are loaded to DS in ECU in CPU Register. Here the DS is Data segment
- The stack segment in RAM are loaded in to the SS that is stack segment.

Step 7

- As the Text, Data, stack segments are stored in CS, SS, DS there are fetch and passed to Next process.
- The ECU contains three more Registers called as ES, FS, GS.

- There are NO full form for ES, FS, GS.
- If the CS, DS and SS are completely full then next segments are stored and fetch in ES, FS, GS. line by line.
- Mostly the Above case does NOT occur.

Step 8

= After the segments are stored then they are fetch and Allied in queue as per the instructions in program. in order of 1, 2, --- n which is step 5 and step 6 in diagram. which is instruction queue.
= The segments are further pass to IP that is Instruction pointer. The Instruction pointer is responsible to fetch the single instruction at a time and pass. Here the IP Act as a Gateway. which is step 6 in the given diagram.

Step 9

- The instructions are pass to CU, ALU and Flags.
- If the instruction related to the Arithmetic then it is passed to the Arithmetic logical unit. otherwise the instruction is passed to the CU that is the control unit.
= And the Flags are responsible to indicate the internal status of the Microprocessor.

Step 10

= Now the instructions get stored inside the CPU registers of RCU
RCU stands for Register control unit.
of the EC then EC stands for Execution unit.
= there are Multiple registers in EC in RCU that are used as per the instruction it completely depends upon the instructions.

Step 11

Here the RCU has several registers and used as per the instruction.

- AH AL
- BH BL
- CH CL
- DH DL
- SP
- BP
- SI
- DI

Here the

- AH, AL are the Arithmatic or Allumation registers
- BH, BL are the Base register
- CH, CL are the count register
- DH, DL are the data register.
- SP is a stack pointer
- BP is a Base pointer
- SI is a source Index
- DI is a destination Index.
- Inside all the above CPU registers the execute of the instructions get performed.

Step 12

- Now the output of instruction get forward to the operating system with the help of BCU and FSB Here the BCU and FSB are
BCU = Bus control unit and
FSB = Front side Bus.

Step 13

- with the help of OS output is displayed on the Moniter / desktop. through the terminal or VS code.

Q what is mean by cache Memory and what are the types of cache Memory ?

→