

LEVEL 02

Machine Level OR Instruction Set Architecture (ISA)

- This level forms the interface between Hardware and Software.
- This makes it possible to execute programs written in different higher level languages. The programs are translated to ISA level and the Hardware can then execute them.

ISA



Tutorialspoint.com
A SIMPLE LEARNING

- This ISA level defines the interface between the Compilers and the hardware. It is language that both of them have to understand.
- The higher level programs are compiled to an ISA program. Then this is executed by Microprogram or hardware.
 - An Instruction Set, or ISA is the part of the Computer Architecture related to programming, including the native Data Types, Instructions, registers, addressing modes, memory architecture, Interrupt and exceptional handling, and external I/O.
 - An ISA includes a specification of the set of opcodes, the native commands implemented by a particular CPU design.
- Example:-** → An operation (ADD) is a part of Instruction set stored in a memory. It is a binary code that tells Computer to perform ADD operation.
- CU decodes instructions from the memory. Then issues a sequence of control signals to initiate microoperations in Internal



LEVEL 03

Operating System OR System Software

- Controls executing processes on the System.
- protects System Resources :- Memory and I/O devices & etc.
- Assembly Language Instructions often pass through Level 3 without modification
- This level is always interpreted

For Instance :- If reading Data from File, the operating System carries it out step by step, in the manner a microprogram would carry out an ADD Instruction.

LEVEL 04

Assembly Language Level

- In this level High level Codes or Language is converted into Assembly Language by a translator called Assembler
- Assembly Language which uses mnemonic Codes is easier than Machine Language using Binary or Hexadecimal Codes.
- It is easier to remember ADD, SUB, MUL or DIV, than their corresponding numerical values in Machine Language
- The assembly language can directly test if there is an overflow bit while a higher level language cannot.
- An Assembly Language can only run on one family of machines (each machine has its own assembly) while a higher level language can run on many machines.

LEVEL 05

High-Level - Language Level

- The level allows users to write their own application with languages such as C, Java and many more.
- High level - languages are easier to read, write and maintain.
- This language is translated into low level language by compilers.

LEVEL 06

The User Level

- program execution and user interface level
- Composed of application programs such as word processor, Paint etc
- The implementation of all six levels below this is hidden from user



TutorialsSpace.com
A SIMPLE LEARNING

Subscribe to our

You Tube Channel