Machine Language Vs Assembly Language Vs High-Level Language

What is Language ?

Language is mode of communication that is used to share ideas, opinions with each other.
For Example : Tamil, English, Mandarin, Japanese, etc..

What is a Programming Language ?

A programming Language is also known as Computer Language, that is used by programmers (developers) to communicate with computers. Simply, Its just a set of instructions written in any specific Language (FORTRAN, C, Java, Python) to perform a specific task.

Types of Programming Language

1. Low-level programming Language

Low-Level language is machine-dependent (0s and 1s) programming Language.

⇒ No compiler or Interpreter
⇒ faster than High-level Language
⇒ Very complicated and tough to understand
⇒ Debugging is complex
⇒ non-portable
⇒ Machine / Architecture independent
⇒ High memory efficient

a) Machine Language

Sometimes called machine code or object code, machine Language is made up of binary numbers or bits that a computer can understand. The exact

machine language for a program or action can differ by the CPU and Operating System, (First generation computers).

For Example :

```
00110100000010000000000000000100
00000001000001010000000001 1000
00000000000000000100000000010010
00000001000001000010000000100000
10001101000010010100000000000000
10001101000010100000000000000100
10101101000010100000000000000000
10101101000010010000000000000100
00000011111100000000000000001000
```

Machine Language (Binary numbers)

```
li   $8, 4          # r8 = 4
mul $8, $8, $4      # r8 = to*i
add $8, $8, $4      # r8 = r8 + base
lw  $9, 0($8)       # r9 = arr[i]
lw  $10, 4($8)      # r10 = arr[i+1]
sw  $10, 0($8)      # arr[i] = r10
sw  $9, 4($8)       # arr[i+1] = r9
jr  $ra             # return
```

MIPS Assembly Language (mnemonic codes)

b) Assembly Language

Assembly Language (ASM) is also a type of low-level programming language that is designed for specific processors. It represents the set of instructions in a symbolic and human-understandble form. It uses an assembler to convert the assembly language to machine language. Components of Assembly Language are Syntax, Label, Operators, Directive, Macro, Mnemonic. It allows direct control over hardware. Unfortunately, It is not portable between machines.

There are two primary types of assemblers.

i) Single-pass assembler

* Scans a program one time → Binary digits (By mnemonic code table)
* often faster than a multi-pass assembler.
* No intermediate code.

ii) multi-pass assembler

* Creates a table with every symbol and each of their values in the first pass, then use the table in future to generate new code.
* bit slower than sigle-pass assembler.
* can be reused for different machines.

For Example:

Example of NASM assembly language code.

```
; program to print "Hello, World"

global  _start

        section .text
_start: mov     rax, 1
        mov     rdi, 1
        mov     rsi, message
        mov     rdx, 13
        syscall
        mov     rax, 60
        xor     rdi, rdi
        syscall

        section .data
message: db     "Hello, World", 10
```

"Hello World" in Java

```
public class Main
{
   public static void main (String [] args) {
         System.out.println ("Hello, World");
   }
}
```

2. High-level programming Language

a) High-Level Languages are programming languages that are designed to allow humans to write computer programs and interact with a computer system without having to have specific knowledge of the processor or hardware that the program will run on. It has "significant abstraction" from the details of computer operation. It may use "natural language elements" and they must be translated by another software called "compiler". Its easy like English. The great acheivement is it can overcome limitation of low-level language. (mainly machine independent). It is faster, less memory efficient, less maintenance, portable, Easily understood by human, etc. The first high-level programming language designed for computers was Plankalkül, created by Konrad Zuse. However, it was not implemented in his time. The first successful high-level language is FORTRAN which is developed by JOHN BACKUS in 1957 with IBM.

For Example,

"Hello, World" in FORTRAN

```
program hello
  print *, 'Hello, World!'
end program hello
```

b) Very High-level programming language (VHLL)

The term VHLL was used in the 1990s for what are today more often called High-level language. A very high-level programming language (VHLL) is a programming language with a very high level of abstraction, used primarily as a professional programmer productivity tool. It includes Automatic programming, OOPS, Feature-oriented programming in well mannered structure. Eg: Java, Javascript, Kotlin, C/C++, Python, Golang, C#, etc..

For example:

"Hello, World" in Mojo Lang (developed in Sep 2022)

```
def main():
print("Hello, World")
```

~ Thank You ~