

Lecture 7

Computer Technology

First Grade

2018-2019

Assistant Prof. Dr. Emad I Abdul Kareem

College of Education

Computer Science Department

الجامعة المستنصرية | Mustansiriyah University



Lecture Seven

Machine language

The native language of the computer, In order for a program to run, it must be presented to the computer as binary-coded machine instructions that are specific to that CPU model or family. Although programmers are sometimes able to modify machine language in order to fix a running program, they do not create it. Machine language is created by programs called "assemblers," "compilers" and "interpreters," which convert the lines of programming code a human writes into the machine language the computer understands. Machine language tells the computer what to do and where to do it.

7.1 Assembly language

Assembly language program using instruction abbreviation called mnemonics, such as LD(load), ST(store) and ADD (Add to Accumulator).This is converted to machine language program with a translator called "assembler".

7.2 High-level-Language (HLL)

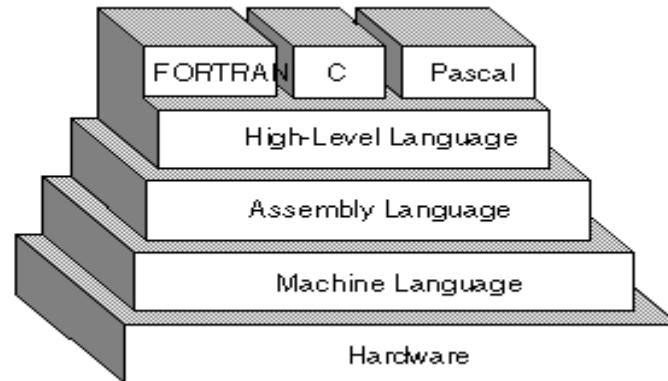
High-Level-programming language is one type of programming Language available. The other type of programming language is known as low-level-language or assembly language. High level-language is easier to learn and understood than the assembly language, because high level languages uses names and commands the resemble English, while the assembly language uses mnemonic codes. Some of the common high-level-languages are:

- FORTRAN (Formula Translation) for engineers.
- -COBOL (Common Business Oriented Language) for business Programmers.
- -Basic (Beginner's All-purpose symbolic Instruction Code) for Engineers and scientists.
- Pascal.

Unlike assembly programs, high-level-languages programs may be used with different makes of computers, while the assembly languages are machine oriented.

Other advantages of high-level-languages are:

- 1) They are easier to learn than assembly languages.
- 2) They are easier to use for problem solving, than assembly.
- 3) They require less time to write, than assembly.
- 4) They provide better documentation.
- 5) They are easier to monition.



7.3 A compiler

Is a program that translates a source language text into a equivalent target language text. What is a compiler?

In order to reduce the complexity of designing and building computers, nearly all of these are made to execute relatively simple commands (but do so very quickly).

A compiler translates (or compiles) a program written in a high-level programming language that is suitable for human programmers into the low-level machine language that is required by computers. During this process, the compiler will also attempt to spot and report obvious programmer mistakes using a high-level language for programming has a large impact on how fast

7.4 Interpreter

Is another way of implementing a programming language. Interpretation shares many aspects with compiling. Lexing, parsing and type-checking are in an interpreter done just as in a compiler. But instead of generating code from the syntax

tree, the syntax tree is processed directly to evaluate expressions and execute statements, and so on. An interpreter may need to process the same piece of the syntax tree (for example, the body of a loop) many times and, hence; interpretation is typically slower than executing a compiled program.

In computer science, an interpreter is a computer program that executes, i.e. *performs*, instructions written in a programming language. An interpreter generally uses one of the following strategies for program execution:

1. Execute the source code directly.
2. translate source code into some efficient intermediate representation and immediately execute this
3. explicitly execute stored precompiled code made by a compiler which is part of the interpreter system