

**Activity #1**

**Student: Angel David Sansores Cruz**

**Tuition: 2109139**

**Subject: Algorithms fundamentals**

**Teacher: Luis Cámara**

**Delivery date: 10/01/2022**

**Activity #1**

**Instructions:** investigate stages of program compilation and levels of programming (description with at least with one example)

**Stages of program compilation**

Compilation is the process of translating source programs into object programs. The object program obtained from the compilation has normally been translated into the common code of the machine language.

To get the real machine program you must use a program called assembler or linker (linker). The assembly process leads to a directly executable machine language program.

The process of running a program written in a programming language and using a compiler typically has the following steps:

1. Write the source program with an editor (program that allows a computer to act similarly to an electronic typewriter) and save it to a storage device (e.g. a disk).
2. Enter the source program in memory.

3. Compile the program with the C compiler.

4. Check and fix compilation errors (list of errors).

5. Obtaining the target program.

6. The linker gets the executable program.

7. The program is executed and, if there are no errors, you will have the output of the program.

Example:

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente (Edukativos, 2013)

**Levels of programming**

In programming there are different types of language. Specialists divide them by levels, from lower to higher complexity. Thus, we find languages of low, medium, and high level.

* **Low-level languages**

They are languages totally dependent on the machine, that is, the program that is made with this type of languages cannot be migrated or used on other machines.

Being practically designed to measure the hardware, they take full advantage of the characteristics of this.

Within this group are:

1. **Machine language:**

This programming language orders the machine the operations fundamental to its operation. It consists of the combination of 0's and 1's (zeros and ones) to form the orders understandable by the hardware of the machine.

This language is much faster than high-level languages.

The disadvantage is that they are quite difficult to handle and use, in addition to having huge source codes where finding a fault is almost impossible.

Example:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. **Assembly language**

This programming language is a derivative of the machine language and is made up of abbreviations of letters and numbers called mnemonics. With the appearance of this language, translator programs were created to be able to pass programs written in assembly language to machine language.

As an advantage over the machine code is that the source codes were shorter, and the programs created occupied less memory.

The disadvantages of this language remain practically the same as those of assembly language, adding the difficulty of having to learn a new language that is difficult to test and maintain.

Example:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente (Mundo”, 2019)

* **Mid-level languages**

These languages are somewhere in between the previous two. Within these languages could be placed "C", since it can access the system registers, work with memory addresses, all of them characteristic of low-level languages and at the same time perform high-level operations.

* **High-level languages**

They are programming languages that are closer to natural language than to machine language. These are languages independent of computer architecture. So, in principle, a program written in a high-level language, you can migrate it from one machine to another without any problem.

These languages allow the programmer to completely forget about the inner workings of the machine for which they are designing the program. They just need a translator who understands the source code as the characteristics of the machine. (Martin, 202o)

Example:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente (Elkan, 2019)

# Bibliografía

*Edukativos*. (13 de agosto de 2013). Obtenido de https://edukativos.com/apuntes/archives/3673

Elkan, M. (22 de julio de 2019). *niveles de programacion* . Obtenido de https://nivelesdeprogramacion.blogspot.com/2019/07/niveles-de-programacion.html

Martin. (08 de junio de 202o). Obtenido de https://catedrauno.com/programacion-lenguajes-niveles-436/

Mundo”, 1. e. (05 de mayo de 2019). Obtenido de https://www.diarlu.com/ejemplos-lenguajes-programacion/