Universidad Politécnica de Yucatán

Homework

Subject: Algorithms Fundamentals

Professor: Luis Gerardo Cámara Salinas

Engineering Program: Data1A

Student: Carlo Alejandro Ek Palomo

Date: 11/ 01/2022

Introduction

The human being has wanted to automate some processes to facilitate some activities, which leads us to the use of applications to carry out these processes but in order to carry out these processes there are certain steps that are followed so that this application works correctly and does what you want is It is necessary to be able to communicate with the system, but the system or the computers only know the binary language, but different languages have been created which makes it easier to communicate with the machines and transmit what you want to achieve or do, tambien existen niveles de

Stages of program compilation

Lexical Analysis.

In this stage, the characters from the source program are read and grouped into strings that represent the lexical components. Each token is a logically coherent sequence of characters relative to an identifier, a reserved word, or a punctuation character. It receives the name of lexeme or its name in English "token" to the sequence of characters that represents a lexical component.

Syntactic analysis.

In this phase, the lexical components are grouped into grammatical phrases that the compiler uses to synthesize the output.

Intermediate code generation.

Some compilers generate an explicit intermediate representation of the source program, after the parse phases have been performed. This intermediate operation can be thought of as a subroutine for an abstract machine. This intermediate representation must have two important properties: it must be easy to produce and easy to translate into the object program.

Code Optimization.

This phase tries to improve the intermediate code, so that it results in machine code that is faster to execute.

Code Generation.

This is the final phase of a compiler. In it the object code is generated, which generally consists of code in machine language (relocatable code) or code in assembly language.

Symbol table manager.

A symbol table is a data structure that contains one record for each identifier. The record includes the fields for the identifier attributes. The symbol table manager is in charge of managing the accesses to the symbol table, in each of the stages of compilation of a program.

Error handler.

At each stage of the compilation process, errors can be found. It is convenient that the error handling is done centrally through an error handler. In this way, the errors found in each of the phases of the compilation of a program can be controlled more efficiently.

There are also different types of levels for programs which will be presented below in which, depending on the level of programming, you are able to understand some languages or build it.

levels of programming

Low level

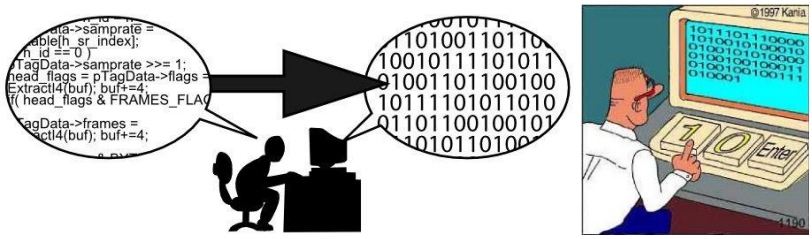
It is one in which its instructions exercise direct control over the hardware, therefore, only the 0 and 1 are used. They are conditioned by the physical structure of the computers that support it.

This level can be classified into 2 types:

Machine language: instructions formed by 0 and 1 that is executed directly in the computer's CPU (processor).

Assembly language: This derivative of machine language, however, uses letters and numbers for instructions.

It is not close to human language and a compiler is necessary to interpret the programmer's instructions.



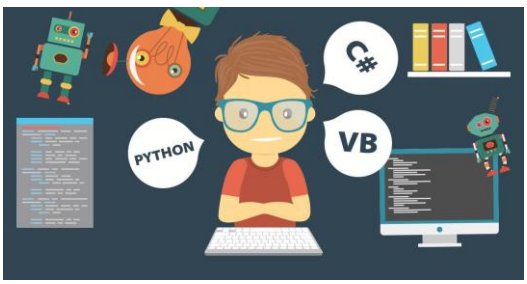
Mid-level languages

They are languages that allow greater abstraction but maintaining some low-level language characteristics. The code is sent to a compiler that converts it to machine language, this language can be placed between low-level and high-level.

High level

It is the one that is closer to human natural language than to the binary language of computers, which is known as low-level language.

Its main function is that, from its development, there is the possibility that the same program can be used on different machines, that is, it is independent of a specific hardware, but it is necessary to have a translator or compiler.



Conclusion

When a computer is required to do a process or an action, during the compilation process, it goes through several phases before performing the action that we normally do not take into account, since we only want it to give us the results we want or perform the action we want, but it is important to know the importance of these phases to better understand the process that a compiler must follow, it is also important to emphasize that there are different levels of programming, to be able to communicate with a system, that the lower your level is closer to a computer language based on binary code and the higher your level is, the language adapts to human language, which is easier for people to understand but also needs more programs so that human language can be transformed into computer language and it can understand the action we want it to perform.

References

Tejada, L .2005. Fases del proceso de compilación. Recovered from https://babotejada.wordpress.com/2007/09/20/fases-de-un-compilador-2/

Alvarez, G.2018.Niveles del lenguajes de programación. Recovered from https://www.kyocode.com/2018/09/niveles-de-lenguajes-de-programacion/