Universidad Politécnica de Yucatán

Subject: Algorithms Fundamentals

Professor: Luis Gerardo Cámara Salinas

Engineering Program: Data

Group: 1-A

Student: Samantha Castro Echeverria

Introduction

Having the theory part clear is one of the most important things, that way we can realize what we did wrong and how to correct it, In this activity two really important topics are ´resented, first we talk about the stages of programming, having clear all the process and all the things that happens while programming is quite important and also interesting to know, the second topic is “levels of programing”, in which we can identify with what kind we as humans can work with and the rest is more for computers.

Stages of program compilation

Compiling is the process of transforming a high-level language into a low-level language. A high-level language is closer to English and low-level language is closer to what the computer understands.

As code is translated, it goes through a few stages to convert the source code from the high-level so we can code that program into machine code so it can be run by the computer.

Many sources failed to agree on an exact number of stages of program compilation, but according to my research I found that there are:

- Lexical analysis
- Syntax analysis
- Semantic analysis
- Code generation

Lexical analysis is also known as a scanner. It converts the High-level input program into a sequence of Tokens.

Syntax analysis is the given input string is checked for the confirmation of rules and structure of the formal grammar. It analyses the syntactical structure and checks if the given input is in the correct syntax of the programming language or not. (Smith, 2021)

Semantic analysis it allows computer to understand and interpret sentences, it makes sure that declarations and statements of program are semantically correct. The grammatical structure is analyzed and relationships between individual words in a particular context are identified.

Code generation is a mechanism where a compiler takes the source code as an input and converts it into machine code.

Levels of programming

Before classifying levels of programing, we have to understand them; so, a programming language defines a set of instructions that are compiled together to perform a specific task by the Central Processing Unit.

There are two main levels in programming

- Low-level language
- High level language

Low-level is a programming language that provides no abstraction from the hardware, and it is represented in 0 or 1 forms an example of this we have the machine level language and the assembly language.

Machine level language consists of a set of instructions that are in the binary form 0 or 1. Example: 01011100

Assembly language is an encoding of machine code into something more readable, it has some human-readable commands such as mov, add, sub, etc.

The high-level languages are considered as high-level because they are closer to human languages than machine-level languages; therefore, make it easier for programmers to think in the programming language. High-level languages also require translation to machine language before execution. Some examples are C, C++, JAVA etc.

Conclusion

In summary, the stages of program compilation are important to know and recognize in order to understand other processes and more complicated things, we saw that it is when we transform the high-level language to low-level language to be able to use it also going into some detail in each stage helped me understand better all this process. Also knowing the levels of programing was very useful, because we might see it as boring theory right now, but it will help us understand upcoming topics.

References

Chawla, M. (2016). *Levels of Programming Languages*. The Bit Theories. Retrieved from: https://thebittheories.com/levels-of-programming-languages-b6a38a68c0f2

Computer Hope. (2019). *What is Assembly Language?* Retrieved from: https://www.computerhope.com/jargon/a/al.htm

Leung, W. (2018). *How the Compilation Process Works for C Programs*. Data Driven Investor. Retrieved from: https://medium.datadriveninvestor.com/compilation-process-db17c3b58e62

Smith, J. (2021). *Syntax Analysis: Compiler Top Down & Bottom Up Parsing Types*. Guru99. Retrieved from: https://www.guru99.com/syntax-analysis-parsing-types.html

Wolff. (2020). *Semantic Analysis, Explained*. Monkey Learn. Retrieved from: https://monkeylearn.com/blog/semantic-analysis/