Sonia Estefanía Mendía Martínez

Data Engineering, 1st Quarter Group: 1A
Student ID: 2109104

Assignment: Algorithms Fundamentals; Stages of compilation & Levels of programming

Due date: 11.01.22

Introduction

In this report we´ll look at 1) the different types of compilation stages, the codes that shape compilation, the respective errors that may happen while coding, and 2) the programming languages that we use nowadays and how they are shape and conform by, as well.

Compilation and its stages

Compilation is a way of turning source code into object code.

What is source and object code?
Source code is a collection of computer instructions written using a human-readable programming language, while Object code is a sequence of statements in machine language, and is the output after the compiler converts the source code.

The compiler checks the source code for language or fundamental errors, and if there are no errors in it, it generates the target code. Translators such as compilers, interpreters, and assemblers are needed to translate programs written in high-level languages into machine code. These tools exist to help programmers develop error-free code.

Stages of compilation

1. lexical análisis (Preprocesssing)
2. symbol table construction
3. syntax analysis
4. semantic analysis
5. code generation
6. optimisation

Lexical analysis	• Comments and unnecessary spaces are removed. • Keywords, constants, and identifiers are replaced with "tokens," which are the symbolic strings used to identify the element.
Symbol table construction	• A table stores the names and addresses of all variables, constants and arrays. • Variables are checked to make sure they have been declared and to determine the data types used.
Syntax analysis	• Tokens are checked to see if they match the syntax of the programming language. • If syntax errors are found, error messages are produced.
Semantic analysis	• Variables are checked to make sure they have been correctly declared and contain the correct data type. • Operations are checked to ensure that they are appropriate for the type of variable being used.
Code generation	• Machine code is generated.
Optimization	• Code optimization makes the program more efficient so it runs faster and uses fewer resources.

Programming errors

There are several types of error that can occur when writing programs:

Syntax	A syntax error is an error in the rules of a sequence of characters that have been written in a particular programming language. For example, punctuation or brackets may be missing, or grammatical symbols such as semicolons or equal signs may be placed in unexpected places
Runtime/Execution	These are errors that occur only during the execution of a program. Errors include running out of memory.
Logical	These occur when the program is running. This can be by using the > symbol or by using AND instead of OR. Logical errors may not prevent programs from running, so they can be hard to find since they are not located on the literal code.
Linking	A linking error occurs when the function or library required by the program cannot be found. This could be because it hasn't been imported or a file has been moved.
Rounding	A rounding error is the difference between the number that is stored by a computer program and the actual number. Data types are typically limited in the number of decimal places they can store, so rounding errors can occur if a number needs to be shortened. An example of this is the number pi (π) which is limited to 3.14 when rounded to two decimal places.
Truncation	Truncation errors occur when reducing a large number to fit a data type. The difference between rounding and truncation is that the extra decimal places are truncated without

	changing the last digit. For example, 5.369 rounded to two decimal places is 5.37, but when truncated to two decimal places it is 5.36.

Programming languages

A programming language defines a set of instructions that are compiled together to perform a specific task by the CPU (Central Processing Unit).

Levels of Programming Language

• Low-level Programming Language

• High-level Programming Language

	Description	Example
Low Level language	Often known as a computer's native language, it provides no abstraction from the hardware, and it is represented in 0 or 1 forms, which are the machine instruction. It is often cryptic. Low-level languages are advantageous because programmes and applications written in them may be executed directly on computer hardware without the requirement for translation or interpretation. Also, these applications and programs can run with a very minimal memory footprint as well as very fast.	Assembly language and machine language are two examples of low-level programming. • FORTRAN, COBOL, BASIC, arguably C Note: machine code is any low-level programming language, Each instruction causes the CPU to perform a very specific task, such as add, subtract, denial, etc.
High Level language	A high-level language is a programming language that allows programmers to write computer-independent programs of a particular type. High-level languages are considered high-level languages because they are	• Python • Java • C++ • C# • Visual Basic

	closer to human languages than machine languages.	• JavaScript

Extra* Assembly language

Assembly language contains human-readable commands such as mov, add, and sub. Assembly language instructions are written in English words such as mov, add, and sub, making them easy to write and understand. It is not portable.

Conclusion

At this point we´ve just learned how we can break up each programming language into their sub-languages such as assembly or machine code. Besides, we are now aware of the multiple errors we can make while coding and where to find them, so we can fix them too. We know how to divide each programming language we know such as Java or C++. into high- or low-level programming languages; this information will help us understand more about its functionality and where we want to apply them.

Biography

Unknown. (2011). What is a Low-Level Language?. 2021, de Javatpoint Sitio web: https://www.javatpoint.com/what-is-a-low-level-language

Unknown . (2011). What is a programming language?. 2021, de Javatpoint Sitio web: https://www.javatpoint.com/classification-of-programming-languages

Gerald Penn. (Unknown). Compilers. 2021, de Toronto Edu Sitio web: https://www.cs.toronto.edu/~gpenn/csc324/lecture2.pdf