



# . STAGES OF PROGRAM COMPILATION

Luis David Martínez Gutiérrez

UNIVERSIDAD POLITECNICA DE YUCATÁN ALGHORTIMS FUNDAMENTALS



## 1. Stages of compilation

The compilation process is a set of stages that source code goes through so that executable object code can be produced.

## 2. Lexical analysis

Lexical analysis is the first stage of the compilation process, where the source code created by the programmer is tokenised for translation into executable code.

The first part of the lexical analysis operation is the removal of any non-program items. Source code is a text document that contains many characters, so it is good programming practice to indent code blocks and use blanks to improve the readability of a program. Additionally, comments can be used to help explain complex parts of the code. These constructs are beneficial to the human reader, but they are not necessary for executable code, so the compiler removes them during lexical analysis.

- Comments and unnecessary spaces are removed.
- Keywords, constants and identifiers are replaced by 'tokens', which are symbolic strings to identify what the elements are.

## 3. Syntax analysis

Syntax analysis is the compilation stage immediately following lexical analysis. Once tokens have been assigned to the code elements, the compiler checks that the tokens are in the correct order and follow the rules of the language. For example, in Python the command `print(user_name)` is syntactically correct, in that it follows the rules for a print statement: print in lowercase immediately followed by a bracket followed by an identifier and closed by a righthand bracket. In abstraction, this is no different from natural languages such as English; the sentence “he Moved wearily” fails on 3 syntax points: the sentence does not start with a capital letter; a capital letter is used incorrectly in the second word; there is no full stop to end the sentence. The syntax rules for programming languages are finite but must be followed.

- Tokens are checked to see if they match the syntax of the programming language.
- If syntax errors are found, error messages are produced.

#### 4. Code generation

This stage follows the stages of lexical and syntax analysis. A separate program is created that is distinct from the original source code. The code generated is the object code, which is the binary equivalent of the source code. This is the executable version of the code, before linked libraries are included.

Code generation is a major distinguishing feature between compilation and interpretation; interpreters do not produce a separate executable file.

- Machine code is generated in this stage.

#### 5. Code optimization

Code optimization is carried out throughout the compilation process and in particular as part of the code generation stage. The optimizer may identify redundant or repeated code, and remove or rearrange the code as necessary. Examples are removing procedures that are never called, or moving an assignment statement that had incorrectly been placed inside a loop, potentially causing it to be inefficiently executed multiple times.

- Code optimization makes the program more efficient so it runs faster and uses fewer resources.

[https://isaaccomputerscience.org/api/v2.19.3/api/images/content/computer\\_science/computer\\_systems/translators/figures/Isaac\\_Computer\\_Science\\_Translators\\_Project\\_V5\\_Artboard\\_1.svg](https://isaaccomputerscience.org/api/v2.19.3/api/images/content/computer_science/computer_systems/translators/figures/Isaac_Computer_Science_Translators_Project_V5_Artboard_1.svg)

<https://www.bbc.co.uk/bitesize/guides/zmthsrd/revision/3>