

Lenguajes de programación

Recuerde que una computadora es un dispositivo electrónico. Las señales eléctricas se utilizan dentro del computadora para procesar la información. Hay dos tipos de señales eléctricas: analógicas y digital. Las **señales analógicas** son formas de ondas continuas que se utilizan para representar cosas como el sonido. Las **señales digitales** representan información con una secuencia de **0** y **1**. Un 0 representa un voltaje bajo y un 1 representa un alto voltaje.

Unit	Symbol	Bits/Bytes
Byte		8 bits
Kilobyte	KB	2^{10} bytes = 1024 bytes
Megabyte	MB	$1024 \text{ KB} = 2^{10} \text{ KB} = 2^{20} \text{ bytes} = 1,048,576 \text{ bytes}$
Gigabyte	GB	$1024 \text{ MB} = 2^{10} \text{ MB} = 2^{30} \text{ bytes} = 1,073,741,824 \text{ bytes}$
Terabyte	TB	$1024 \text{ GB} = 2^{10} \text{ GB} = 2^{40} \text{ bytes} = 1,099,511,627,776 \text{ bytes}$
Petabyte	PB	$1024 \text{ TB} = 2^{10} \text{ TB} = 2^{50} \text{ bytes} = 1,125,899,906,842,624 \text{ bytes}$
Exabyte	EB	$1024 \text{ PB} = 2^{10} \text{ PB} = 2^{60} \text{ bytes} = 1,152,921,504,606,846,976 \text{ bytes}$
Zettabyte	ZB	$1024 \text{ EB} = 2^{10} \text{ EB} = 2^{70} \text{ bytes} = 1,180,591,620,717,411,303,424 \text{ bytes}$

¹ Yottabyte (YT) 2^{80} bytes

La evolución de los lenguajes de programación

Los lenguajes de programación se clasifican en:

- **Máquina:** Código máquina.

El lenguaje más básico de una computadora, el **lenguaje de máquina**, proporciona programas instrucciones en bits. Aunque la mayoría de las computadoras realizan el mismo tipo de operaciones, los diseñadores de la computadora pueden haber elegido diferentes conjuntos de códigos binarios para realizar las operaciones. Por lo tanto, el lenguaje de máquina de una máquina no es necesariamente el igual que el lenguaje de máquina de otra máquina. La única coherencia entre las computadoras es que, en cualquier computadora moderna, todos los datos se almacenan y manipulan como códigos binarios.

Ejemplo:

Para calcular los salarios semanales. Además, suponga que el código binario 100100 significa carga, 100110 significa multiplicación y 100010 significa tienda. En lenguaje de máquina, podría necesitar la siguiente secuencia de instrucciones para calcular los salarios semanales:

100100	010001
100110	010010
100010	010011

Para representar la ecuación de salario semanal en lenguaje de máquina, el programador tuvo que recordar los códigos de lenguaje de máquina para varias operaciones. Además, para manipular los datos, el programador tenía que recordar las ubicaciones de los datos en la memoria principal.

Esta necesidad de recordar códigos específicos hizo que la programación no solo fuera muy difícil, sino también propenso a errores.

- **Bajo nivel:** Ensamblador.

Los **lenguajes ensambladores** se desarrollaron para facilitar el trabajo del programador. En el lenguaje ensamblador, una instrucción es una forma fácil de recordar a una llamada *mnemotécnica*.

Ejemplo:

Assembly Language	Machine Language
LOAD	100100
STOR	100010
MULT	100110
ADD	100101
SUB	100011

Usando instrucciones en lenguaje ensamblador, puede escribir la ecuación para calcular el semanario salarios de la siguiente manera:

LOAD *rate*

MULT *hours*

STOR *wages*

Como puede ver, es mucho más fácil escribir instrucciones en lenguaje ensamblador. Sin embargo, una computadora no puede ejecutar instrucciones en lenguaje ensamblador directamente. Las instrucciones primero tienen que ser traducidos al lenguaje de máquina. Un programa llamado **assembler** traduce las instrucciones en lenguaje ensamblador a lenguaje de máquina.

- **Alto nivel:** Pascal, FORTRAN, Visual Basic, C, Ada, Modula-2, C++, Java, Delphi, C#, etc.

Pasar del lenguaje máquina al lenguaje ensamblador facilitó la programación, pero un programador todavía se veía obligado a pensar en términos de instrucciones de máquina individuales. El siguiente paso para facilitar la programación fue diseñar **lenguajes de alto nivel** que estaban más cerca de los lenguajes naturales, como inglés, francés, alemán y español.

Ejemplo: En C ++

*wages = rate * hours;*

- **Diseño de Web:** SMGL, HTML, XML, PHP...

El proceso de traducción de un programa fuente escrito en un lenguaje de alto nivel a un lenguaje máquina comprensible por la computadora, se realiza mediante programas llamados “traductores”. Los traductores de lenguaje son programas que traducen a su vez los programas fuente escritos en lenguajes de alto nivel a código máquina. Los traductores se dividen en **compiladores** e **intérpretes**.

- Intérpretes

Un intérprete es un traductor que toma un programa fuente, lo traduce y, a continuación, lo ejecuta. Los programas intérpretes clásicos como BASIC, prácticamente ya no se utilizan, más que en circunstancias especiales. Sin embargo, está muy extendida la versión interpretada del lenguaje Smalltalk, un lenguaje orientado a objetos puro. El sistema de traducción consiste en:

Traducir la primera sentencia del programa a lenguaje máquina, se detiene la traducción, se ejecuta la sentencia; a continuación, se traduce la siguiente sentencia, se detiene la traducción, se ejecuta la sentencia y así sucesivamente hasta terminar el programa.

- Compiladores

Un compilador es un programa que traduce los programas fuente escritos en lenguaje de alto nivel a lenguaje máquina. La traducción del programa completo se realiza en una sola operación denominada compilación del programa; es decir, se traducen todas las instrucciones del programa en un solo bloque. El programa compilado y depurado (eliminados los errores del código fuente) se denomina programa ejecutable porque ya se puede ejecutar directamente y cuantas veces se desee; sólo deberá volver a compilarse de nuevo en el caso de que se modifique alguna instrucción del programa. De este modo el programa ejecutable no necesita del compilador para su ejecución. Los traductores de lenguajes típicos más utilizados son: C, C++, Java, C#, Pascal, FORTRAN y COBOL.

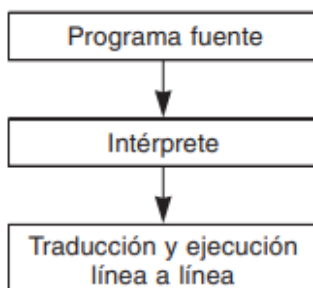


Figura 1.16. Intérprete.

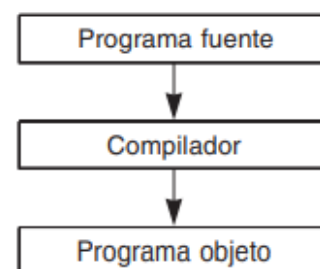


Figura 1.17. La compilación de programas.

La compilación y sus fases:

La compilación es el proceso de traducción de programas fuente a programas objeto. El programa objeto obtenido de la compilación ha sido traducido normalmente a código máquina.

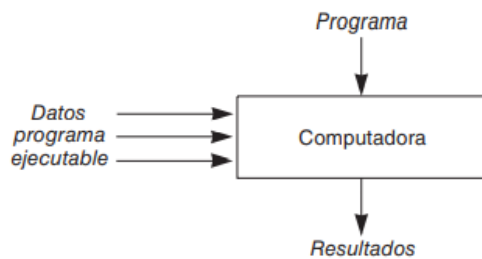


Figura 1.19. Ejecución de un programa.

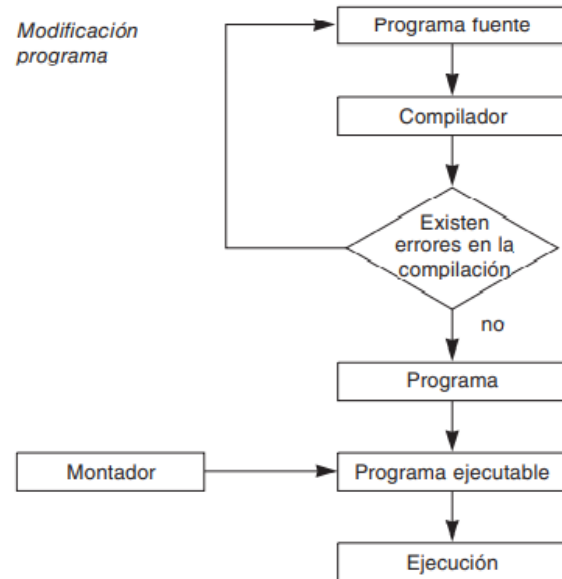


Figura 1.20. Fases de ejecución de un programa.