

# **Fundamentos del lenguaje ensamblador**

## **UNIDAD 1**

# ENSAMBLADOR

Es un lenguaje de programación de bajo nivel para los computadores, microprocesadores, microcontroladores y otros circuitos integrados programables.

Implementa una representación simbólica de los códigos de máquina binarios y otras constantes necesarias para programar una arquitectura dada de CPU y constituye la representación más directa del código máquina específico para cada arquitectura legible por un programador

# OBJETIVOS DEL TEMA

- Conocer los bloques funcionales que forman el *hardware* de un sistema de cómputo
- Entender cómo ejecuta un computador *realmente* un programa o *software*
- Aprender a realizar una interacción correcta entre *hardware* y *software*

Los objetivos se logran aprendiendo el Lenguaje Ensamblador (Assembly), único lenguaje que permite programar el hardware de la máquina DIRECTAMENTE.

Los otros lenguajes se convierten en traductores que ocultan al programador los detalles internos del sistema físico.

Surge la interrogante:

*¿Para qué aprender Ensamblador?*

- Para entender el funcionamiento interno, a nivel electrónico, de un sistema de cómputo
- Para programar tareas críticas de los sistemas operativos, de aplicaciones en tiempo real o controladores de dispositivos.
- Para entender lo que realmente ocurre cuando se ejecuta un programa.

# TIPOS DE LENGUAJES



LENGUAJE DE ALTO NIVEL  
(JAVA, C#, PHP, PYTHON, ETC)

LENGUAJE DE BAJO NIVEL  
ENSAMBLADOR

CÓDIGO MÁQUINA  
(BINARIO [0-1])

HARDWARE  
(PARTE FISICA DE LA COMPUTADORA)

# TIPOS DE LENGUAJES

Alto nivel

Cobol  
Pascal  
Basic  
Java

Perl C#  
Python RUBY ON RAILS  
PHP C  
Java AJAX JavaScript  
VB. Net

Mediano nivel

C  
BCPL

Bajo nivel

Ensamblador

Máquina

001110011010100

Instrucciones que ejecuta el microprocesador una tras otra. Primero las busca en la memoria y después las ejecuta.

Confeccionar programas

son

sirven para

## Lenguajes de programación

- Cobol
- Fortran
- C, C++
- Java
- C #

pueden ser

algunos son

Compiladores

necesitan

Intérpretes

Lenguaje máquina

alto nivel

ventajas

inconvenientes

- Menos complicados de programar.
- Sólo es necesario conocer las instrucciones del lenguaje.

- No los interpreta directamente el ordenador.

inconvenientes

Difíciles de confeccionar.  
Complicados para el usuario.

se usa

Circuitos internos

bajo nivel

algunos son

Lenguaje  
Ensamblador

necesitan

traductor

Ensamblador

ventajas

Los que mejor "entiende" el microprocesador.  
Se parecen al lenguaje máquina.  
Son muy rápidos.

Lenguaje máquina



Un lenguaje de alto nivel permite al programador escribir las instrucciones de un programa utilizando palabras o expresiones sintácticas muy similares al inglés.

## Lenguajes de alto nivel

- \* Ada
- \* ALGOL
- \* Basic
- \* C++
- \* Clipper
- \* COBOL
- \* Fortran
- \* Java
- \* Modula-2
- \* Pascal
- \* PHP
- \* PL/SQL
- \* Perl
- \* Prolog
- \* Python
- \* Ruby
- \* Lenguajes funcionales
  - Haskell
  - Lisp

# Lenguajes de medio nivel

Un lenguaje de medio nivel (como es el caso del *lenguaje C*) tiene ciertas características que lo acerca a los lenguajes de bajo nivel: pueden acceder a registros de memoria, y al mismo tiempo, tiene ciertas cualidades que lo hacen un lenguaje más cercano al humano y, por tanto, de alto nivel.

*Los lenguajes de bajo nivel son mas fáciles de utilizar que los lenguajes máquina, pero, al igual que ellos, dependen de la máquina en particular.*

*El lenguaje de bajo nivel por excelencia es el ensamblador.*

#### Lenguaje de máquina

- ▶ Serie de 0's y 1's
- ▶ Programación larga, difícil y tediosa
- ▶ La corrección de errores es complicada

#### Lenguaje ensamblador

- ▶ Utiliza una serie de códigos o mnemónicos
- ▶ Especifico de cada procesador
- ▶ Difícil aprendizaje

- *Las instrucciones en lenguaje ensamblador son instrucciones conocidas como nemotécnicos.*

*Por ejemplo, nemotécnicos típicos de operaciones aritméticas son: en inglés, ADD, SUB, DIV, etc.; en español, SUM, RES, DIV, etc. Una instrucción típica de suma sería:*

*ADD M, N, P*

*Esta instrucción podría significar "sumar el número contenido en la posición de memoria M al número almacenado en la posición de memoria N y situar el resultado en la posición de memoria P".*

*Evidentemente es mucho más sencillo recordar la instrucción anterior con un nemotécnico que su equivalente en código máquina".*

*0110 1001 1010 1011*

*Las instrucciones en lenguaje ensamblador (nemotécnicos), que utilizaremos son como estas:*

```
NOP
PUSH EBX
PUSH ESI
PUSH EDI
MOV ESI,EDX
MOV EBX,EAX
TEST ESI,ESI
MOV EDI,EBX
JE SHORT RTRACE.004012EA
PUSH 4
PUSH 1000
PUSH 1000
PUSH EBX
CALL <JMP.&KERNEL32.VirtualAlloc>
TEST EAX,EAX
JNZ SHORT RTRACE.004012DA
MOV EDX,EBX
MOV EAX,EDI
SUB EDX,EDI
CALL RTRACE.004012F4
XOR EAX,EAX
JMP SHORT RTRACE.004012EF
ADD EBX,1000
SUB ESI,1000
TEST ESI,ESI
JNZ SHORT RTRACE.004012B5
MOV EAX,1
POP EDI
POP ESI
POP EBX
RETN
NOP
```

# TRADUCTORES DE LOS LENGUAJES

Los traductores de lenguaje son los programas que traducen a su vez los programas fuente escritos en lenguajes de alto nivel a código máquina. Los cuales son:

**Compilador:** Lenguaje de alto nivel -> Lenguaje Máquina

**Intérprete:** Un traductor que toma un programa fuente, lo traduce y a continuación lo ejecuta.



# TRADUCTORES DE LOS LENGUAJES

## COMPILADORES

Los traductores de tipo compilador convierten el código fuente de un programa en código máquina, generando en el camino un recurso adicional (un ejecutable) que será el que utilizaremos para ejecutar el programa.

En realidad, el programa fuente (programa con el código fuente, valga la redundancia) será convertido, sentencia a sentencia, a código máquina, creando un programa objeto o código objeto.

Sin embargo para crear el programa final, autoejecutable, será necesario un proceso adicional: el enlazado o montaje (realizado por el programa montador, enlazador o linker).

El resultado final, será un programa autoejecutable que no necesitará ninguna herramienta adicional para hacerlo. Simplemente le indicaremos que se ejecute cuando lo deseemos y así lo hará.

# TRADUCTORES DE LOS LENGUAJES

## Intérprete

Es un traductor que ejecuta las líneas de código que conforman un programa una a una y directamente. Es un programa que va leyendo el código fuente de otro programa y lo va ejecutando según lo lee.

El código máquina generado en el momento de la traducción no es almacenado para su posterior ejecución cuando se solicite. Por tanto, el programa realizado con un lenguaje de programación traducido por un intérprete, siempre necesitará a éste para poder ejecutarse. No existe independencia entre la fase de traducción y ejecución.

Cada vez que el programa necesite ejecutarse, el traductor tendrá que convertir el código fuente a código máquina (interpretar). Esta característica hace que, por regla general, los lenguajes que usan este tipo de traductor ofrezcan un menor rendimiento.

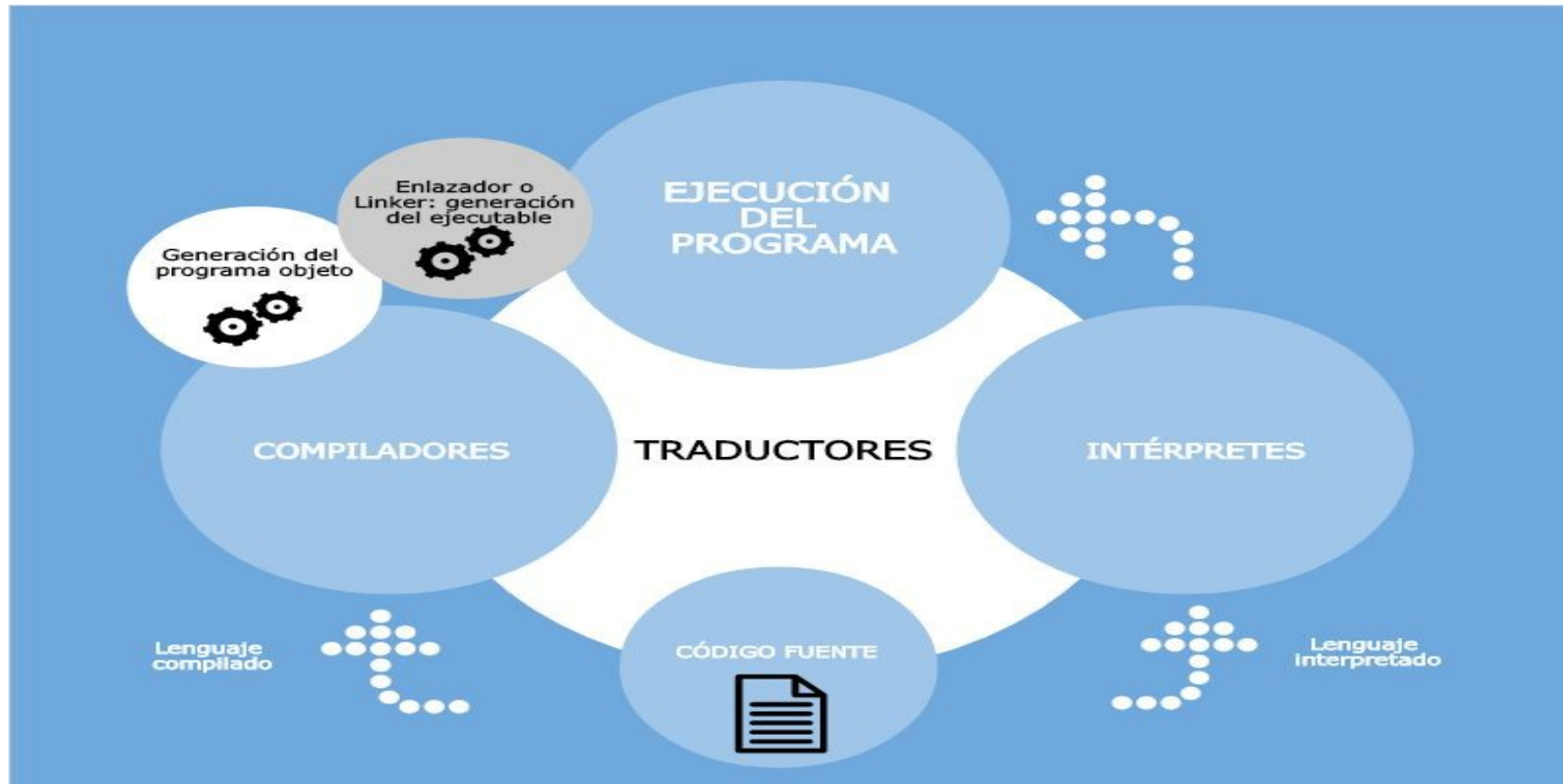
En contrapartida, ofrecen un mejor entorno de programación, depuración y mantenimiento que el otro conjunto de traductores: los compiladores.



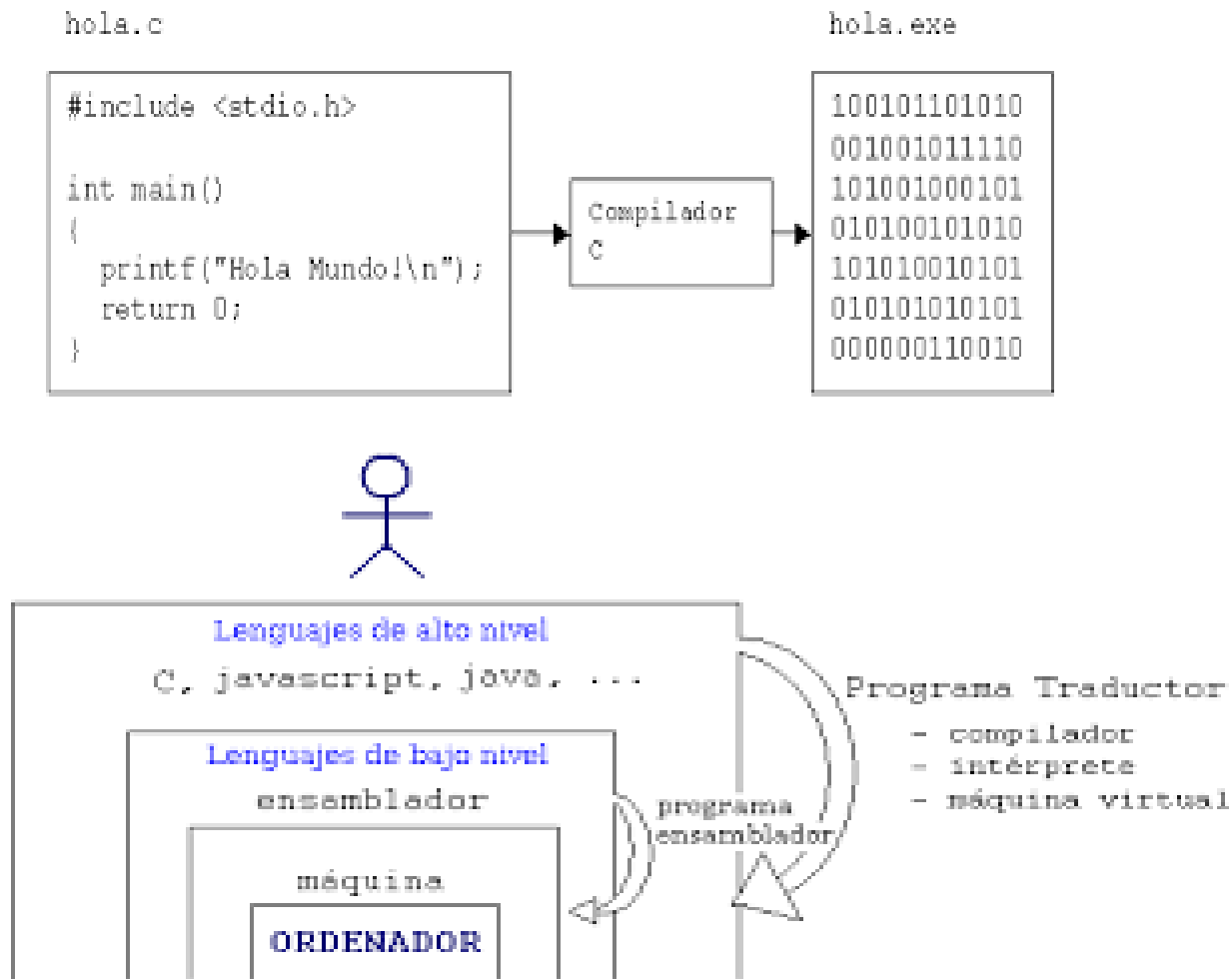
# TRADUCTORES DE LOS LENGUAJES

Existen otros traductores que producen un código intermedio entre el código fuente y el máquina. Son por tanto, una **mezcla de interpretados y compilados**.

En el lenguaje Java por ejemplo, se genera un recurso bytecode: código precompilado que necesita interpretarse por la JVM o máquina virtual de Java para ejecutarse.



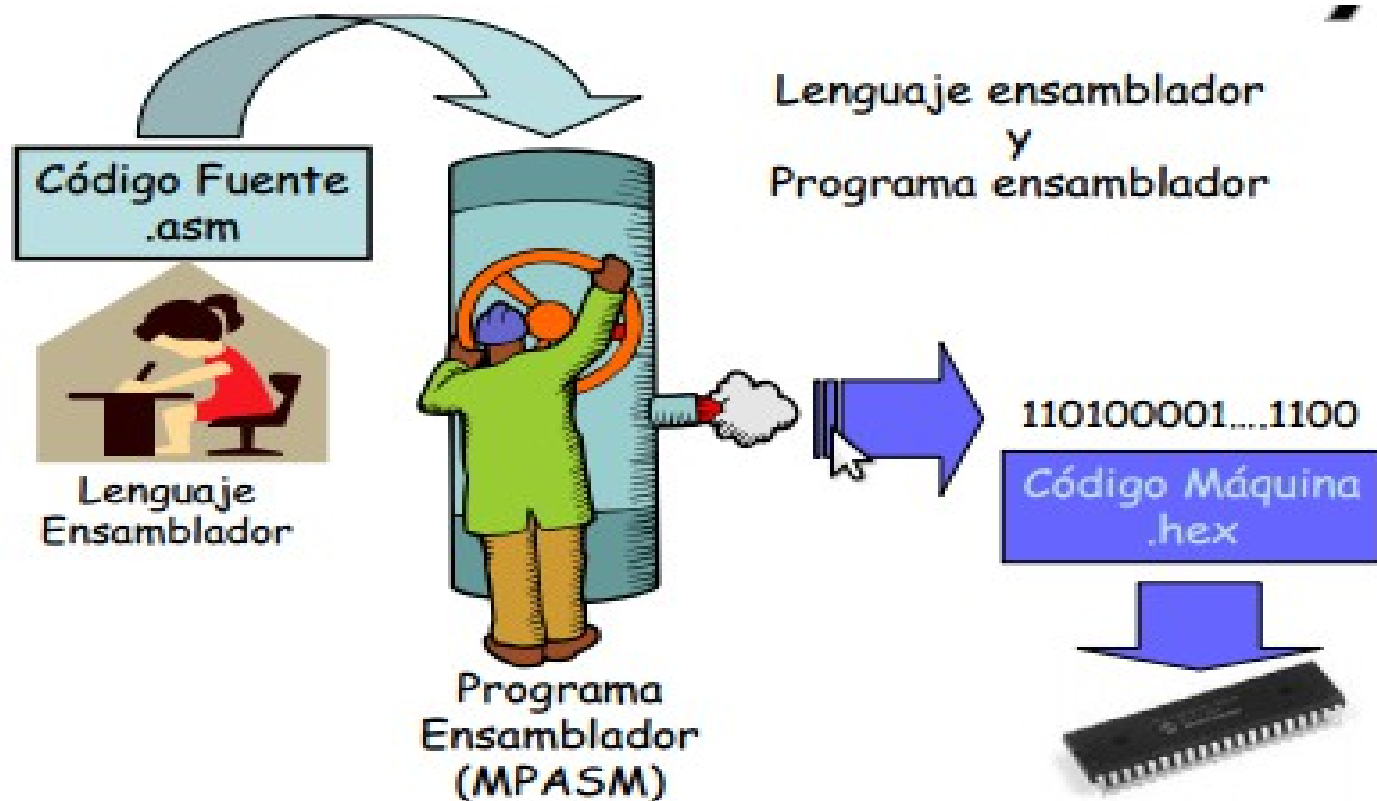
# TRADUCTORES DE LOS LENGUAJES



# TRADUCCIÓN DEL LENGUAJE ENSAMBLADOR

- *Un programa escrito en lenguaje ensamblador no puede ser ejecutado directamente por la computadora en esto se diferencia esencialmente del lenguaje máquina, sino que requiere una fase de traducción al lenguaje máquina.*
- *El programa original escrito en lenguaje ensamblador se denomina programa fuente y el programa traducido en lenguaje máquina se conoce como programa objeto, ya directamente entendible por la computadora.*

- *El traductor de programas fuente a objeto es un programa llamado ensamblador, existente en casi todos los computadores.*



# ***Ventajas del lenguaje ensamblador.***

- *Los lenguajes ensambladores presentan la ventaja frente a los lenguajes máquina de su mayor facilidad de codificación.*
- *En general, su velocidad de cálculo.*
- *El control total de la PC que se tiene con el uso del mismo.*

# ***Desventajas del lenguaje ensamblador.***

- *Dependencia total de la máquina lo que impide la transportabilidad de los programas. El lenguaje ensamblador del PC es distinto del lenguaje ensamblador del Apple Macintosh.*
- *La formación de los programadores es más compleja que la correspondiente a los programadores de alto nivel, ya que no sólo las técnicas de programación, sino también el conocimiento del interior de la máquina.*