


Assembly es un lenguaje de programación de bajo nivel que se desarrolló en la década de 1940. Su historia está intrínsecamente ligada al surgimiento de las primeras computadoras electrónicas. En sus primeras encarnaciones, los programadores tenían que escribir instrucciones en código máquina directamente, lo que era extremadamente tedioso y propenso a errores. Para hacerlo más accesible, se desarrolló el lenguaje Assembly, que permitía a los programadores escribir códigos más humanamente legibles, pero aún muy cercanos a la arquitectura de la máquina.

Principales Características de Assembly:

1. **Bajo Nivel:** Assembly se encuentra muy cerca del lenguaje de máquina y proporciona un alto grado de control sobre el hardware de la computadora. Los programas escritos en Assembly son específicos de la arquitectura del procesador.
2. **Instrucciones Simples:** Las instrucciones Assembly suelen ser simples y representan operaciones fundamentales directamente comprensibles para el hardware. Estas operaciones incluyen cargar valores en registros, realizar cálculos aritméticos y controlar el flujo del programa.
3. **Eficiencia:** Dado que los programas Assembly se ejecutan directamente en el hardware, son altamente eficientes en términos de velocidad y uso de recursos. Esto los hace adecuados para aplicaciones que requieren un rendimiento extremadamente rápido.
4. **Dificultad:** Es más difícil de aprender y escribir en comparación con lenguajes de alto nivel debido a su nivel de abstracción más bajo y a la necesidad de conocer la arquitectura específica del procesador.

Ejemplo: Linux



```
1 section .text
2     global _start    ;must be declared for linker (ld)
3     _start:          ;tells linker entry point
4         mov edi,len   ;message length
5         mov ecx,msg    ;message to write
6         mov ebx,1      ;file descriptor (stdout)
7         mov eax,4       ;system call number (sys_
8         int 0x80        ;call kernel
9
10        mov eax,1       ;system call number (sys_
11        int 0x80        ;call kernel
12
13 section .data
14 msg db 'Hello, world!', 0xa ;string to be pr
15 len equ $ - msg        ;length of the string
```

☒ Result:

Compiling main.asm source code....
Snasm -f elf main.asm 2>&1

wikiHow

El lenguaje de programación C fue creado por Dennis Ritchie en los laboratorios Bell de AT&T en la década de 1970. Ritchie desarrolló C como una evolución del lenguaje de programación B, que a su vez estaba basado en el lenguaje BCPL. C se convirtió en un lenguaje de programación ampliamente influyente y se usó inicialmente para el desarrollo del sistema operativo Unix. A medida que Unix ganó popularidad, C también lo hizo, y su portabilidad y eficiencia lo llevaron a ser ampliamente adoptado en una variedad de aplicaciones y sistemas.

Principales Características de C:

1. **Portabilidad:** C se diseñó para ser altamente portátil, lo que significa que el código escrito en C se puede compilar y ejecutar en una variedad de plataformas y arquitecturas de hardware con relativamente pocas modificaciones.
2. **Eficiencia:** C permite un control cercano del hardware y la memoria, lo que lo hace ideal para la programación de sistemas y aplicaciones que requieren un alto rendimiento.
3. **Sintaxis Clara y Concisa:** La sintaxis de C es relativamente simple y legible, lo que facilita la escritura y el mantenimiento del código.
4. **Librerías Estándar:** C incluye una amplia variedad de funciones y bibliotecas estándar que simplifican el desarrollo de software. Esto significa que los programadores pueden aprovechar las bibliotecas existentes para tareas comunes.
5. **Punteros:** C es conocido por su uso extensivo de punteros, lo que permite un control preciso de la memoria. Si bien esto puede ser poderoso, también puede ser propenso a errores si no se maneja correctamente.

Unreal Engine y Unity utilizan estos lenguajes, y muchos juegos populares, como World of Warcraft y Counter-Strike



```
0100000000010100011011000000100101100011
110001011101000100011111111110100000100
001010010110000110101110110110110010001
0110110000010101100100010000111000100111
0100110010110100110110100111101111011110
000110100#include <stdio.h>01101000011010
0100100110001010001110
1000100int main()000101111
010101001f00011000
111001100printf("Hello World")0001100
001000001return 42;010101110110
000110100010001100011000110100011010
0100100110111101011101110000001010001110
100010010001010110010011101110100010111
0101010011100110101011100010101010001100
111001100000110111111010100111110001100
0100000111111101010010010011010101110110
```

Python es un lenguaje de programación de alto nivel que fue creado por Guido van Rossum y lanzado por primera vez en 1991. La creación de Python se inspiró en la idea de tener un lenguaje de programación que fuera fácil de leer y escribir, con una sintaxis clara y un enfoque en la legibilidad del código. El nombre "Python" proviene de la afición de van Rossum por los humoristas británicos Monty Python.

A lo largo de los años, Python ha experimentado varias versiones, siendo Python 2 y Python 3 las más notables. Python 3 introdujo mejoras significativas en la sintaxis y la funcionalidad, aunque también generó algunas incompatibilidades con Python 2. En la actualidad, Python 3 es la versión recomendada y ampliamente utilizada.

Principales Características de Python:

1. **Sintaxis Clara y Legible:** Python se enorgullece de su sintaxis fácil de leer, que utiliza sangrías en lugar de llaves o paréntesis para definir bloques de código. Esto hace que el código sea más legible y reduce la propensión a cometer errores de formato.
2. **Lenguaje Versátil:** Python es un lenguaje multipropósito que se puede utilizar en una amplia variedad de aplicaciones, desde desarrollo web hasta análisis de datos, aprendizaje automático, automatización, scripting y más.
3. **Gran Comunidad y Ecosistema:** Python tiene una comunidad activa y una gran cantidad de bibliotecas y módulos disponibles. Esto permite a los desarrolladores aprovechar soluciones preexistentes para una amplia gama de tareas.
4. **Portabilidad:** Python es compatible con una variedad de sistemas operativos, lo que lo hace altamente portátil. Los programas escritos en Python pueden ejecutarse en diferentes plataformas sin necesidad de modificaciones significativas.
5. **Interpretado:** Python es un lenguaje interpretado, lo que significa que no es necesario compilar el código antes de ejecutarlo. Esto facilita el desarrollo y la depuración rápidos.
6. **Programación Orientada a Objetos:** Python admite programación orientada a objetos (POO) y proporciona una forma elegante de definir y trabajar con objetos y clases.

Ejemplo: YouTube



```
304     endless_query = self.  
305     endless_query.start_position = self.start_position  
306     self.all_patches = self.compute_percentile(percentage)  
307     return self.all_patches  
308  
309     def compute_percentile(self, percentage):  
310         """  
311         Returns a Position object that represents a percentage-far-of-the-way  
312         through the larger task, as specified by this query.  
313         """  
314         @param percentage: a number between 0 and 100.  
315         all_patches = self.get_all_patches(cache=False)  
316         return all_patches  
317  
318     def generate_patches(self):  
319         """  
320         Generates a list of patches that satisfy the given conditions given  
321         """  
322         self.root_directory = self.get_root_directory()  
323         self.all_patches = self.get_all_patches(cache=False)  
324         self.all_patches = self.get_all_patches(cache=False)
```

1. Nivel de Abstracción:

- **Assembly:** Es un lenguaje de bajo nivel que está estrechamente relacionado con la arquitectura del hardware de la computadora. Los programas escritos en ensamblador se componen de instrucciones específicas de la CPU y son difíciles de leer y escribir para los humanos.
- **C:** Es un lenguaje de nivel medio que proporciona un nivel de abstracción más alto que el ensamblador, pero aún permite un control detallado sobre el hardware. Es más legible que el ensamblador, pero menos abstracto que Python.
- **Python:** Es un lenguaje de alto nivel que se centra en la facilidad de lectura y escritura. Tiene una abstracción significativamente mayor que el ensamblador y C, lo que lo hace más fácil de aprender y usar para los programadores.

2. Portabilidad:

- **Assembly:** El código en ensamblador es altamente dependiente de la arquitectura del procesador. Los programas escritos en ensamblador no son portables y deben ajustarse para funcionar en diferentes plataformas.
- **C:** C es más portátil que el ensamblador, ya que se ha diseñado para ser independiente de la plataforma. Los programas en C generalmente se pueden compilar en diferentes sistemas con modificaciones mínimas.
- **Python:** Python es altamente portátil y se ejecuta en una variedad de sistemas operativos y arquitecturas sin necesidad de modificaciones significativas en el código fuente.

3. Velocidad de Ejecución:

- **Assembly:** Los programas escritos en ensamblador a menudo se ejecutan muy rápido, ya que están cerca del nivel del hardware. Sin embargo, escribir en ensamblador puede ser lento y propenso a errores.
- **C:** Los programas en C son generalmente rápidos debido a su proximidad al hardware, pero también ofrecen un equilibrio entre velocidad y facilidad de programación.
- **Python:** Python es más lento en términos de velocidad de ejecución en comparación con Assembly y C, debido a su alto nivel de abstracción y gestión de memoria automática. Sin embargo, es más rápido de desarrollar y depurar.

4. Uso y Dominio:

- **Assembly:** Se utiliza principalmente para programación de sistemas, control de hardware y optimización de código de bajo nivel.
- **C:** Es ampliamente utilizado en el desarrollo de sistemas operativos, controladores de dispositivos, aplicaciones de alto rendimiento y juegos.
- **Python:** Se utiliza en una variedad de aplicaciones, como desarrollo web, análisis de datos, inteligencia artificial, automatización y scripting debido a su facilidad de uso y legibilidad.