Logotipo, nombre de la empresa

Descripción generada automáticamente

Proyecto Investigacion Final

Paradigmas de la Programación

**Integrantes:**

Anders Flores Morales   
Alvaro Chaves Cascante

Investigación acerca de

FORTH

**Profesor:**

Josías Chaves Murillo

**Curso:**  
Paradigmas de la Programación

18 octubre 2023

**Introducción**

El lenguaje de programación Forth es un lenguaje poco convencional, que se caracteriza por ser procedimental, estructurado, imperativo, reflexivo, basado en pila y sin comprobación de tipos. Fue creado por Charles H. [Moore entre los años 1965 y 1970 en el Observatorio Nacional de Radioastronomía de Kitt Peak, Arizona1](https://es.wikipedia.org/wiki/Forth). [Su nombre es una contracción de la palabra inglesa fourth, que significa cuarto, ya que sus creadores le consideraban destinado a la cuarta generación de computadoras1](https://es.wikipedia.org/wiki/Forth). Sin embargo, el nombre se quedó en FORTH debido a las limitaciones de longitud de los nombres en el IBM 1130, el primer ordenador en el que se implementó el lenguaje[1](https://es.wikipedia.org/wiki/Forth). [Forth es un lenguaje que ofrece tanto la ejecución interactiva de comandos como la capacidad de compilar secuencias de comandos para la ejecución posterior1](https://es.wikipedia.org/wiki/Forth). [Algunas de sus aplicaciones más destacadas son los cargadores (boot loaders), las aplicaciones espaciales y los sistemas empotrados1](https://es.wikipedia.org/wiki/Forth).

**Antecedentes**

El origen del lenguaje Forth se remonta a la década de 1960, cuando Charles H. [Moore trabajaba como astrónomo en el Observatorio Nacional de Radioastronomía de Kitt Peak, Arizona1](https://es.wikipedia.org/wiki/Forth). [Moore se interesó por la programación y empezó a experimentar con diferentes lenguajes, como ALGOL, COBOL y LISP2](https://www.nachocabanes.com/forth/). [Sin embargo, ninguno de ellos le satisfacía plenamente, ya que le parecían demasiado complejos, lentos o ineficientes para sus necesidades2](https://www.nachocabanes.com/forth/). [Por ello, decidió crear su propio lenguaje, que fuera simple, rápido y flexible2](https://www.nachocabanes.com/forth/). [Así nació Forth, un lenguaje basado en una pila de datos y en la definición de palabras o subrutinas que se pueden invocar con un solo nombre2](https://www.nachocabanes.com/forth/). Moore usó Forth para realizar cálculos astronómicos, como el análisis de espectros de emisión o el cálculo de trayectorias de cuerpos en órbita[1](https://es.wikipedia.org/wiki/Forth). [Posteriormente, un programa para la adquisición automática y continua de datos realizado en este lenguaje descubrió al menos la mitad de los cúmulos interestelares conocidos en la actualidad1](https://es.wikipedia.org/wiki/Forth).

**Análisis del tema**

Forth es un lenguaje de programación que se puede clasificar como procedimental, estructurado, imperativo, reflexivo, basado en pila y sin comprobación de tipos. A continuación se explican estas características:

* **Procedimental:** significa que el programa se compone de una serie de procedimientos o subrutinas que realizan una tarea específica. Estos procedimientos se llaman palabras en Forth y se definen con las palabras clave : y ;. Por ejemplo, la palabra SUMA que suma dos números se define así:

: SUMA + . ;

* **Estructurado**: significa que el programa se organiza en bloques o estructuras que controlan el flujo de ejecución. Estas estructuras pueden ser condicionales (IF…ELSE…THEN), iterativas (DO…LOOP) o recursivas (RECURSE). Por ejemplo, la palabra FACTORIAL que calcula el factorial de un número se define así:

: FACTORIAL DUP 1 = IF DROP 1 ELSE DUP 1 - RECURSE \* THEN ;

* **Imperativo**: significa que el programa se ejecuta mediante una secuencia de instrucciones que modifican el estado del sistema. Estas instrucciones pueden ser operaciones aritméticas (+ - \* /), lógicas (AND OR NOT), comparativas (= < >), manipulación de pila (DUP DROP SWAP OVER), entrada/salida (. ." KEY EMIT), etc. Por ejemplo, para mostrar el resultado de sumar 4 y 7 se escribe:

4 7 SUMA

* **Reflexivo**: significa que el programa puede modificar su propio código fuente o crear nuevas palabras durante la ejecución. Esto se logra mediante las palabras CREATE y DOES>, que permiten definir palabras con datos asociados y comportamiento dinámico. Por ejemplo, la palabra VARIABLE crea una palabra con un espacio reservado para almacenar un valor:

: VARIABLE CREATE 0 , DOES> @ ;

* **Basado en pila**: significa que el programa usa una pila de datos para pasar los argumentos entre las palabras. La pila es una estructura de datos de tipo LIFO (Last In First Out), es decir, el último elemento en entrar es el primero en salir. Las palabras pueden apilar o desapilar elementos de la pila según su número de parámetros y de resultados. Por ejemplo, la palabra + apila dos números y desapila su suma:

4 7 + ( apila 4, apila 7, desapila 4 y 7, apila 11 )

* **Sin comprobación de tipos**: significa que el programa no verifica el tipo de los datos que maneja. Esto implica que el programador debe asegurarse de que los datos son coherentes con las operaciones que se realizan sobre ellos. Por ejemplo, si se intenta sumar un número y una cadena de texto, se produce un error:

4 "Hola" + ( error: tipos incompatibles )

**Características generales son:**

Es un lenguaje de bajo nivel, es decir, que está cerca del funcionamiento de la máquina y permite un control directo del hardware.

Es un lenguaje extensible, es decir, que permite al programador definir nuevas palabras o modificar las existentes según sus necesidades.

Es un lenguaje conciso, es decir, que usa pocas palabras o símbolos para expresar las instrucciones.

Es un lenguaje interactivo, es decir, que permite al programador ejecutar y depurar el código en tiempo real, sin necesidad de compilar o reiniciar el programa.

**Características específicas son:**

Es un lenguaje polaco inverso, es decir, que usa la notación postfija para escribir las expresiones.

Es un lenguaje basado en pilas, es decir, que usa una estructura de datos llamada pila para almacenar y manipular los valores.

Es un lenguaje basado en palabras, es decir, que usa palabras como unidades básicas de programación. Una palabra es una secuencia de caracteres que representa una acción o un valor. Las palabras pueden ser primitivas (definidas por el sistema) o compuestas (definidas por el usuario). Por ejemplo, la palabra . (punto) es una palabra primitiva que imprime el tope de la pila y lo elimina. La palabra DUP es una palabra compuesta que duplica el tope de la pila. Se puede definir así: : DUP @ + ; donde : indica el inicio de la definición, @ indica el tope de la pila y ; indica el final de la definición.

**Ventajas**

Forth es un lenguaje de programación que tiene las siguientes ventajas:

* Es un lenguaje eficiente, es decir, que consume pocos recursos de memoria y procesamiento. Esto lo hace ideal para programar sistemas con limitaciones de hardware o rendimiento.
* Es un lenguaje portable, es decir, que se puede ejecutar en diferentes plataformas o arquitecturas. Esto lo hace compatible con diversos sistemas operativos o dispositivos.
* Es un lenguaje flexible, es decir, que se puede adaptar a diferentes paradigmas o estilos de programación. Esto lo hace versátil y creativo.

**Desventajas**

Forth es un lenguaje de programación que tiene las siguientes desventajas:

* Es un lenguaje difícil, es decir, que requiere una curva de aprendizaje elevada y una buena comprensión de la pila y los tipos de datos. Esto lo hace complejo y propenso a errores.
* Es un lenguaje poco usado, es decir, que tiene una escasa difusión y aceptación entre los programadores. Esto lo hace marginal y poco documentado.
* Es un lenguaje poco estandarizado, es decir, que tiene varias implementaciones o versiones con diferencias significativas. Esto lo hace inconsistente e incompatible.

**Futuro del lenguaje**

Forth es un lenguaje de programación que tiene un futuro incierto o básicamente inexistente, ya que depende del interés y el apoyo de la comunidad de usuarios y desarrolladores. Algunas posibles tendencias son:

* Mantener el espíritu original del lenguaje, preservando su simplicidad, rapidez y extensibilidad. Esto implicaría seguir usando Forth para aplicaciones específicas o nichos de mercado, pero por su forma de uso es ineficiente y no hay interés por lo que se podría decir que no tiene futuro ya.

**Conclusiones**

Forth es un lenguaje de programación único y original, que se distingue por su diseño basado en una pila de datos y en la definición de palabras o subrutinas. Fue creado por Charles H. Moore en la década de 1960, con el objetivo de simplificar y acelerar la programación de sistemas astronómicos. Desde entonces, ha sido usado para diversas aplicaciones, especialmente en el campo de los sistemas empotrados, como los cargadores, las aplicaciones espaciales o los dispositivos electrónicos.

Forth tiene ventajas como su eficiencia, su portabilidad, su flexibilidad y su interactividad, que le permiten adaptarse a diferentes plataformas, dominios y estilos de programación. Sin embargo, también tiene desventajas como su dificultad, su escaso uso y su poca estandarización, que le hacen complejo, marginal e inconsistente.

Forth tiene un futuro incierto, ya que depende del interés y el apoyo de la comunidad de usuarios y desarrolladores. Algunas posibles tendencias son mantener el espíritu original del lenguaje, evolucionar el lenguaje hacia nuevas características o funcionalidades o integrar el lenguaje con otros sistemas o herramientas.

**Recomendaciones**

Para aprender más sobre el lenguaje de programación Forth, se recomienda consultar las siguientes fuentes:

* El libro [Starting Forth], de Leo Brodie, que es una introducción clásica y amena al lenguaje.
* El sitio web [Forth.com], que es el portal oficial del lenguaje y ofrece recursos, noticias y enlaces sobre Forth.
* El sitio web [ForthHub], que es una plataforma colaborativa para compartir proyectos, código y documentación sobre Forth.

**Bibliografía**

: Brodie, L. (1981). Starting Forth. Prentice-Hall. : Forth.com. (2023).

The Forth Programming Language. Retrieved from [https://www.forth.com/] : ForthHub. (2023).

ForthHub: A place for all things Forth. Retrieved from [https://github.com/ForthHub/ForthHub]

https://es.wikipedia.org/wiki/Forth

**Prácticas:**

* Para practicar el lenguaje Forth, puedes usar un intérprete o un compilador que te permita ejecutar y depurar el código en tiempo real. Algunas opciones son [gforth], [SwiftForth] o [Win32Forth], que son implementaciones gratuitas y multiplataforma de Forth. También puedes usar un entorno de desarrollo integrado (IDE) que te facilite la edición y el manejo de los archivos de código. Algunas opciones son [ForthIDE], [ForthPad] o [ForthBuilder], que son herramientas específicas para Forth. Otra forma de practicar el lenguaje Forth es participar en desafíos o concursos de programación, como el [IOCCC] o el [Code Golf Stack Exchange], que te retan a crear programas cortos y originales en Forth.

**Ejemplos:**

* Para ver ejemplos de programas escritos en Forth, se puede consultar los siguientes recursos:
  + El libro [Starting Forth], de Leo Brodie, que contiene numerosos ejemplos y ejercicios para aprender el lenguaje desde cero.
  + El sitio web [Rosetta Code], que es un repositorio de soluciones a problemas comunes de programación en diferentes lenguajes, incluyendo Forth.
  + El sitio web [ForthHub], que es una plataforma colaborativa para compartir proyectos, código y documentación sobre Forth.

también algunos ejemplos sencillos son:

* Al ser polaco inverso implica que los operadores se escriben después de los operandos, sin necesidad de paréntesis. Por ejemplo, la expresión (4 + 7) \* 2 se escribe así: 4 7 + 2 \*
* Al tener ser un lenguaje también de tipo pila funciona como una lista donde solo se puede acceder al último elemento (el tope de la pila) y donde se pueden realizar operaciones de apilar (push) y desapilar (pop). Por ejemplo, la expresión anterior se evalúa así: 4 7 + 2 \* -> 4 (push) -> 4 7 (push) -> 4 11 (pop + push) -> 4 11 2 (push) -> 22 (pop \* push).

**Funcionamiento:**

* Para entender el funcionamiento del lenguaje Forth, debes conocer sus principales características, que son las siguientes:
  + Es un lenguaje procedimental, estructurado, imperativo, reflexivo, basado en pila y sin comprobación de tipos.
  + Es un lenguaje de bajo nivel, extensible, conciso e interactivo.
  + Es un lenguaje polaco inverso, que usa la notación postfija para escribir las expresiones.
  + Estas características se explican con más detalle en la sección anterior del ensayo sobre el lenguaje Forth.
* **Instalación:**
* Para instalar el lenguaje Forth en tu computadora, debes seguir los siguientes pasos:
  + Elegir una implementación o versión de Forth que sea compatible con tu sistema operativo y arquitectura. Algunas opciones son [gforth], [SwiftForth] o [Win32Forth], que son implementaciones gratuitas y multiplataforma de Forth.
  + Descargar el archivo ejecutable o comprimido desde el sitio web oficial o desde una fuente confiable. Por ejemplo, descargar gforth para Windows 10, se puede usar este enlace: [gforth-0.7.3\_20191024-setup.exe].
  + Ejecutar el archivo descargado y seguir las instrucciones del instalador. Por ejemplo, para instalar gforth para Windows 10, puedes hacer clic en Next, aceptar los términos del acuerdo de licencia, elegir la carpeta de destino y completar la instalación.
  + Verificar que la instalación se haya realizado correctamente y que puedas acceder al intérprete o al compilador de Forth desde la línea de comandos o desde el menú de inicio. Por ejemplo, para verificar gforth para Windows 10, puedes abrir una ventana del símbolo del sistema y escribir gforth. Si todo está bien, deberías ver algo así: