



ANTECEDENTES Y DEFINICIÓN DE LENGUAJE C

Fundamentos de Programación

Peñaloza Castañeda Luna Quetzalli

Código de estudiante: 221618438

Maestra: Sánchez Rosario Patricia

Horario: martes y jueves 11:00 am – 12:55 pm

07/09/2024

NRC: 200274

Historia del lenguaje C

El lenguaje C tiene una estrecha relación con el sistema operativo UNIX. Creado entre 1972 y 1973, ha influido en la creación muchísimos otros lenguajes de programación que conocemos en la actualidad.

Su historia comienza con el desarrollo de Unix en 1969 en Bell Labs. Originalmente, Unix se programaba en lenguaje ensamblador: un lenguaje de bajo nivel por las limitaciones de la época y la nula existencia de lenguajes de alto nivel. Esto representaba un desafío en términos de portabilidad y eficiencia. Ken Thompson, uno de los principales desarrolladores de UNIX se encontraba buscando una solución par que el desarrollo de aplicaciones en esta nueva plataforma se simplificara.

Fue aquí cuando Thompson creó un lenguaje simplificado basado en BCPL (Basic Combined Programming Language) llamado B. B poseía un compilador que traducía el código fuente al código máquina, facilitando de gran manera la portabilidad del sistema. A pesar de todo esto, B resultó ser extremadamente lento y limitado en funcionalidad.

Años después, en 1971 Dennis Ritchie (otro miembro del grupo del desarrollo de Unix) junto con Brian Kernighan comenzaron a mejorar B debido a la necesidad de un lenguaje más potente y eficiente debido a que UNIX necesitaba un lenguaje que pudiera aprovechar al máximo las capacidades del hardware, permitiendo la escritura de un código eficiente y portable. Esta nueva versión fue originalmente conocida como “nuevo B” o por sus siglas “NB”, e introdujo nuevas funcionalidades como tipos de datos y punteros a memoria. Al ver que NB cumplía con los requisitos al permitir la manipulación directa de hardware y memoria, Thompson comenzó a usar NB en lugar del ensamblador para programar UNIX y sus necesidades fueron modelando el desarrollo del lenguaje. Se creó un nuevo compilador y el lenguaje fue creativamente renombrado como “C”.

La versión 2 de UNIX ya incluía el compilador de C y algunas utilidades, y la versión 4 lanzada en noviembre de 1973 fue reimplementada casi por completo en C, favoreciendo de gran manera su portabilidad a otras maquinas. También incluyó varias mejores y clarificaciones, como la definición de la biblioteca estándar y más importante aún: la sintaxis del lenguaje. Este estándar estableció una base sólida para que el lenguaje se siguiera desarrollando y se adoptara en una gran variedad de plataformas y aplicaciones.

Dejando de lado cualidades del lenguaje en sí por un momento, un paso importante en su difusión y lo que modeló la industria informática por completo fue la publicación del libro “The C Programming Language” escrito por los mismísimos Ritchie y Kernighan (¡El autor del primer “Hola Mundo” conocido!). Esto fue importantísimo, pues era una época donde apenas existía información sobre informática y muchas veces era de boca en boca. Esta obra sirvió como una

especificación informal y un estándar que cualquiera que quisiera escribir un compilador podría utilizar.

En 1980 surge el lenguaje C++, una extensión orientada a objetos del lenguaje C y desarrollada por Bjarne Stroustrup en los laboratorios de At&T.

En 1983 se creó un comité de estandarización en ANSI (American National Standards Institute) para definir un estándar para el lenguaje C conocido como ANSI C.

En 1985 apareció el sistema operativo Windows, y tanto Windows como Mac Os (dos de los sistemas operativos mas importantes en la era actual) fueron escritos en C.

Entre 1989 el estándar ANSI C fue publicado oficialmente como ANSI X3.159-1989 o comúnmente conocido como C89 y en 1990 fue adoptado por la ISO (International Organization for Standardization). Esta estandarización fue crucial para consolidar el lenguaje y garantizar que los programas escritos en C fueran portables entre diferentes sistemas, y desde entonces, las distintas versiones de C se han publicado como un estándar ampliamente aceptado.

En 1991 se lanza el sistema operativo Linux, la cual podría llamarse como una de las contribuciones más significativas en el mundo del software libre y de código abierto, y se desarrolló precisamente utilizando el lenguaje de programación C. Este sistema operativo es actualmente una pieza fundamental de la infraestructura tecnológica moderna, y como dato curioso, alrededor del 97% de las 500 supercomputadoras más potentes del mundo entero utilizan el núcleo de Linux, lo que demuestra su capacidad y fiabilidad en entornos de alta demanda.

Pero el lenguaje nunca dejó de evolucionar, y en 1999 se publica el estándar C99, que introdujo varias mejoras como tipos de datos nuevos (por ejemplo, "long long", que son tipos de datos enteros largos de 64 bit, permitiendo un rango mayor en las operaciones aritméticas), soporte para comentarios en una sola línea (//), y mejoras en la biblioteca estándar, incluyendo soporte para operaciones de matemáticas complejas y funciones de manipulación de cadenas.

En el año 2000, Microsoft desarrolla el lenguaje C#, un "descendiente" del lenguaje C. C# fue diseñado principalmente para el desarrollo de aplicaciones .NET de Microsoft, combinando elementos y características tanto de C como de Java.

Años después se introducen los estándares C11 y C18, donde C11 (Publicado en diciembre del 2011) introduce varias mejoras con respecto a C99, como mejoras en la seguridad para evitar desbordamientos de búfer, compatibilidad con Unicode, entre otros. Por su lado, C18 es una actualización menor publicada en julio de 2018, y se considera más que nada una corrección de errores y mejora de C11, sin introducir cambios significativos en el lenguaje.

A partir de aquí el resto es historia. C Sigue siendo un lenguaje relevante y altamente utilizado. Las nuevas versiones del estándar están desarrollándose cuidadosamente para equilibrar la modernización y la compatibilidad. Me resulta crucial recalcar que antes de C ya existían lenguajes de alto nivel como FORTRAN o COBOL que tenían cierta portabilidad, pero tenían un dominio funcional específico y además eran bastante ineficientes en el uso de memoria y tiempo de ejecución. Ritchie desarrolló C como una solución a este problema, diseñándolo para ser las dos cosas que no se tenían en ese tiempo: portable y eficiente. C no solo ha sido fundamental en el desarrollo de sistemas operativos como UNIX, sino que también ha influido en la creación de muchos otros lenguajes de programación modernos además de los mencionados C++ y C#, como JavaScript y Python en la actualidad.

Definición

Una vez sabiendo su historia, podemos entender que el lenguaje C es un lenguaje de programación que se utiliza para desarrollar software de todo tipo, desde sistemas operativos hasta aplicaciones de usuario.

También es necesario mencionar su sintaxis, ya que es relativamente simple, facilitando su aprendizaje y la escritura correcta del código. Esta flexibilidad y capacidad de manipular directamente la memoria hacen que sea potente y eficiente.

Además, una característica clave de C es su minimalismo, pues fue pensado como una herramienta sencilla para escribir programas que se compilan con un compilador, es decir, el código original se convierte en un archivo que las máquinas pueden leer directamente sin aplicaciones adicionales de por medio.

Es un lenguaje C es de nivel medio-bajo, por lo que no requiere de muchos recursos y ofrece una buena respuesta y ejecución rápida de programas.

Hablando sobre su sintaxis, empezaremos con los tokens C. Estos son considerados como la unidad más pequeña de programación en este lenguaje, y son palabras clave, identificadores, constantes, variables o cualquier otro símbolo que tenga algún significado en el lenguaje C.

Podría decirse que un programa escrito en C es como un “juego de tokens”, y la sintaxis determina como es que se combinan estos tokens para escribir el código.

Cada bloque de código se encierra en corchetes, y la base de cada bloque es la función principal `main()`. Otra cosa de gran importancia es que al final de cada operador se tiene que poner un punto y coma, y si se te olvida ponerlo el compilador mostrará un mensaje de error.

Otra de sus características distintivas fue su implementación nativa del paradigma de programación estructurada, introduciendo conceptos como subrutinas y estructuras condicionales (“*if*” y “*while*”) y de iteración (bucles *for* y *while*). Estas abstracciones dieron paso a los programadores a escribir un código más legible y

mantenible. **C impulsó de forma decisiva la adopción de la programación estructurada en toda la industria.**

Bibliografía

1. Bonilla, D. (2023, septiembre 6). *La Historia del lenguaje C*. La Voz de Galicia. <https://www.lavozdeg Galicia.es/noticia/reto-digital/ocio/2023/09/05/historia-lenguaje-c/00031693928294754171450.htm> Recuperado el 7 de septiembre de 2024.
2. Cuadrado, G. C. (2019, septiembre 4). Qué es C: Todo lo que debes saber. *Openwebinars.net*. <https://openwebinars.net/blog/que-es-c/> Recuperado el 7 de septiembre de 2024.
3. de un programa en lenguaje C, E. P. T. se D. a. D. la E., De datos, la D., & la entrada/salida básica que permitirán hacer los primeros programas., las I. M. S. y. la C. de E. Y. se T. C. las S. P. R. (s/f). *Programación en (ANSI) C: Primeros pasos*. Unican.es. Recuperado el 8 de septiembre de 2024, de <https://ocw.unican.es/pluginfile.php/2336/course/section/2291/T-Uno.pdf>
4. Saavedra, J. A. (2023, marzo 22). *Qué es lenguaje C: el origen, las ventajas, las características y la sintaxis del lenguaje de programación*. Ebac. <https://ebac.mx/blog/que-es-lenguaje-c> Recuperado el 8 de septiembre de 2024.
5. Vieito, E. (2024, mayo 7). *Historia del lenguaje de programación C: ¿por qué sigue estando tan presente?* El blog de dinahosting; Dinahosting. <https://dinahosting.com/blog/historia-del-lenguaje-de-programacion-c-por-que-sigue-estando-tan-presente/> Recuperado el 8 de septiembre de 2024.