

Resumen Lenguaje de programación C

Introducción

El lenguaje de programación C fue creado por Brian Kernaghan y Dennis Ritchie a mediados de los años 70, y es un lenguaje estructurado, pero no por bloques. Es decir que no es posible declarar pequeños trozos del programa.

Uno de los objetivos del diseño del lenguaje C es que solo necesita unas pocas instrucciones en el lenguaje para traducir cada elemento del lenguaje sub que tenga un gran soporte en el tiempo de ejecución.

El lenguaje C, con la popularidad de las microcomputadoras, generó diversas versiones, lo que llevó a que ANSI estableciera en 1983 un comité para definir un estándar universal y no dependiente de la máquina.

Es un lenguaje de nivel medio pues combina ciertas características de un lenguaje de alto nivel y el ensamblador, pero algunas de sus principales cualidades son que tiene una alta portabilidad y que tiene una programación estructurada mediante módulos o funciones. Aunque fue creado para sistemas operativos hoy en día lo utilizamos para todo tipo de programas.

Elementos generales.

Todos los lenguajes tienen características comunes pero los elementos de un programa en C son:

Tiene comentarios que se escriben entre `/* */` y documentan el programa sin afectar su ejecución.

Incluyen archivos los cuales permiten llamar a bibliotecas donde están definidas funciones estándar necesarias.

La función principal es `main()`, pues marca el inicio del programa y es delimitado con llaves{
}

Las variables locales deben declararse antes de utilizarlas en operaciones dentro del programa

Su conjunto de instrucciones son los flujos de sentencias y forman el cuerpo del programa

Se crean funciones definidas por el programador para ser llamadas desde `main()` t contienen la misma estructura básica.

Para ejecutarlo hay que

Oprimir la tecla F10 y seleccionar la opción File.Posiciones en Open

Escribir en el nombre del archivo donde va a ir guardado el programa.
Automáticamente se le agrega la terminación .c

Características y propiedades

Un núcleo del lenguaje simple, con funcionalidades añadidas importantes, como:

Flexibilidad la cual permite distintos estilos de programación, destacando el estructurado.

Sistema de tipos que se encarga de evitar operaciones sin sentido.

Preprocesador el cual es usado para macros e inclusión de archivos.

Punteros los cuales permiten acceso directo a la memoria.

Uniones que posibilitan interrupciones al procesador.

Palabras clave reducidas

Paso de parámetros es decir que son ordenados por valor, y por referencia usando direcciones de memoria.

Punteros a funciones y variables estáticas las cuales son las que brindan encapsulación y polimorfismo básicos.

Y finalmente los Structs que agrupan datos relacionados en una sola variable.

El lenguaje C, a pesar de ser uno de los más utilizados y apreciados por su rapidez y eficiencia, presenta también algunas carencias importantes que lo diferencian de otros lenguajes más modernos.

Una de ellas es la ausencia de un sistema de recolección de basura nativo, es decir, C no administra automáticamente la memoria que ya no se utiliza.

Esto obliga al programador a encargarse de liberar la memoria manualmente, aunque existen bibliotecas como libgc de Sun Microsystems o el recolector de basura de Boehm que pueden añadirse para suplir esta limitación.

Otra carencia es el soporte para programación orientada a objetos. C fue diseñado como un lenguaje estructurado, no orientado a objetos, aunque posteriormente se desarrolló C++, que en sus inicios era un preprocesador capaz de traducir el código de C++ a C.

El lenguaje tampoco permite, en su versión estándar, la definición de funciones anidadas. Sin embargo, algunos compiladores como GCC ofrecen esta posibilidad como una extensión propia.

Del mismo modo, C carece de un soporte nativo para programación multihilo, lo cual significa que el programador debe recurrir a bibliotecas externas o a soluciones específicas de cada sistema operativo para manejar varios procesos en paralelo.

A pesar de que la lista de características faltantes puede parecer larga, esto ha jugado un papel positivo en su aceptación, ya que hace que sea más fácil crear compiladores para nuevas plataformas y que el programador tenga un control directo sobre lo que realmente sucede en el programa.

Incluso algunas de las limitaciones de C pueden resolverse de forma creativa.

Por ejemplo, en C++ se utiliza la notación stack.push(val) para trabajar con objetos y métodos.

En cambio, en C se puede escribir algo equivalente como push(stack, val) donde el argumento stack es un puntero a una estructura que cumple la misma función que el puntero this en C + +.

Tipos de variables en C

Para la definición de variables se debe poner primero la variable definida y después el nombre de la variable.

Si queremos definir más de una variable del mismo tipo podemos hacerlo colocando una coma.

Por ejemplo

```
int contador; /* define una variable con el nombre contador de tipo entero */
```

```
int cont = 0 ; /* inicializando cont con 0 */
```

```
int cont , cont2, contn = 0; /* definición de mas de una variable */
```

```
float científica = 10.1 int cont = cont +1;
```

```
int cont = cont2
```

Motivos para escoger este texto científico

En lo personal, me parece interesante conocer cómo funcionan los lenguajes de programación, pero sobre todo C, pues personalmente no es un lenguaje el cual domino, pero me gustaría conocerlo.

Me parece importante conocer la importancia y el uso de cada lenguaje de programación, pues dependiendo de los objetivos algunos son más útiles que otros, pero otro aspecto que me parece relevante es que aprender un lenguaje de programación te puede llevar a conocer y aprender otros lenguajes con mayor habilidad, pues muchos comparten una lógica similar.

También me motiva a conocer más elementos que desconocía del lenguaje C y que lo hacen un lenguaje especial, como que es un lenguaje eficiente o su estructura, y es un lenguaje muy utilizado.