

Lenguaje de Programación C

Introducción.

El lenguaje de programación C o lenguaje de programación de sistemas ofrece una economía sintáctica, control de flujo y estructuras sencillas y un buen conjunto de operadores. Este fue creado por Brian Kernaghan y Dennis Ritchie a mediados de los años 70.

El lenguaje C es un lenguaje estructurado, en el mismo sentido que lo son otros lenguajes de programación tales como el Pascal, el Ada o el Modula-2, pero no es estructurado por bloques, o sea, no es posible declarar subrutinas (pequeños trozos de programa) dentro de otras subrutinas, a diferencia de como sucede con otros lenguajes de estructurados como Pascal.

Uno de los objetivos de diseño del lenguaje C es que sólo sean necesarias unas pocas intrucciones en lenguaje de máquina para traducir cada elemento del lenguaje, sin que haga falta un soporte intenso en tiempo de ejecución.

Con la popularidad de las microcomputadoras muchas compañías comenzaron a implementar su propio C por lo cual surgieron discrepancias entre sí.

Por esta razón ANSI (American National Standards Institute, por sus siglas en inglés), estableció un comité en 1983 para crear una definición no ambigua del lenguaje C e independiente de la máquina que pudiera utilizarse en todos los tipos de C.

Algunos de las C existentes son:

- Quick C
- C++
- Turbo C
- Turbo C++
- Borland C
- Borland C++
- Microsoft C
- etc.

Este manual se basa en el C estandar el cual puede utilizarse en todos los tipos de C.

C es un lenguaje de programación de nivel medio

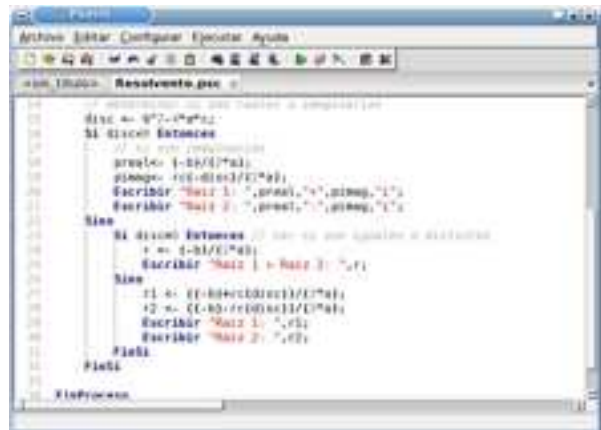
ya que combina los elementos del lenguaje de alto nivel con la funcionalidad del ensamblador.

Su característica principal es ser portable, es decir, es posible adaptar los programas escritos para un tipo de computadora en otra.

Otra de sus características principales es el ser estructurado, es decir, el programa se divide en módulos (funciones) independientes entre sí.

El lenguaje C inicialmente fué creado para la programación de

- Sistemas operativos



- Intérpretes
- Editores
- Ensambladores
- Compiladores
- Administradores de bases de datos.

Actualmente, debido a sus características, puede ser utilizado paratodo tipo de programas.

Elementos generales.

Aunque cada uno de los programas son distintos, todos tienen características comunes. Los elementos de un programa en C son los siguientes:

Comentarios

Inclusión de archivos

main()

{

variables locales

flujo de sentencias

}

Definición de funciones creadas por el programador utilizadas en main()

Veamos en que consiste cada uno:

Comentarios: Se identifican porque van entre diagonales y asterisco. Nos sirve para escribir información que nos referencie al programa pero que no forme parte de él. Por ejemplo especificar que hace el programa, quien lo elaboró, en que fecha, que versión es, etc.

Inclusión de archivos: Consiste en mandar llamar a la o las bibliotecas donde se encuentran definidas las funciones de C (instrucciones) que estamos utilizando en el programa.

En realidad, la inclusión de archivos no forma parte de la estructura propia de un programa sino que pertenece al desarrollo integrado de C. Se incluye aquí para que el alumno no olvide que debe mandar llamar a los archivos donde se encuentran definidas las funciones estandar que va a utilizar.

main(): En C, todo está constituido a base de funciones. El programa principal no es la excepción.

main() indica el comienzo de la función principal del programa la cual se delimita con llaves.

Variables locales: Antes de realizar alguna operación en el programa, se deben declarar la(s) variable(s) que se utilizarán en el programa.

Flujo de sentencias: Es la declaración de todas las instrucciones que conforman el programa.

Definición de funciones creadas por el programador utilizadas en main(): Finalmente, se procede a definir el contenido de las funciones utilizadas dentro de main(). Estas contienen los mismos elementos que la función principal.

Para ejecutarlo, realice los siguientes pasos:

* Oprima la tecla F10 y seleccione la opción File. Posicione en Open y pulse enter.

* Escriba el nombre del archivo donde va a guardar su programa, que no exceda de ocho letras.

Automáticamente se le agregará la terminación .c. Digite <ENTER>.

* Transcriba el programa.

Leer más: <http://www.monografias.com/trabajos/introc/introc.shtml#ixzz3nltuxfs7> Estructura básica de un programa en C

Características

Propiedades

- Un núcleo del lenguaje simple, con funcionalidades añadidas importantes, como funciones matemáticas y de manejo de archivos, proporcionadas por bibliotecas.
- Es un lenguaje muy flexible que permite programar con múltiples estilos. Uno de los más empleados es el estructurado "no llevado al extremo" (permitiendo ciertas licencias de ruptura).
- Un sistema de tipos que impide operaciones sin sentido.
- Usa un lenguaje de preprocesado, el preprocesador de C, para tareas como definir macros e incluir múltiples archivos de código fuente.
- Acceso a memoria de bajo nivel mediante el uso de punteros.
- Interrupciones al procesador con uniones.
- Un conjunto reducido de palabras clave.
- Por defecto, el paso de parámetros a una función se realiza por valor. El paso por referencia se consigue pasando explícitamente a las funciones las direcciones de memoria de dichos parámetros.
- Punteros a funciones y variables estáticas, que permiten una forma rudimentaria de encapsulado y polimorfismo.
- Tipos de datos agregados (`struct`) que permiten que datos relacionados (como un empleado, que tiene un id, un nombre y un salario) se combinen y se manipulen como un todo (en una única variable "empleado").

Carencias

- Recolección de basura nativa, sin embargo se encuentran a tal efecto bibliotecas como la "libc" desarrollada por Sun Microsystems, o el Recolector de basura de Boehm.
- Soporte para programación orientada a objetos, aunque la implementación original de C++ fue un preprocesador que traducía código fuente de C++ a C.
- Funciones anidadas, aunque GCC tiene esta característica como extensión.
- Soporte nativo para programación multihilo.

Aunque la lista de las características útiles de las que carece C es larga, este factor ha sido importante para su aceptación, porque escribir rápidamente nuevos compiladores para nuevas plataformas, mantiene lo que realmente hace el programa bajo el control directo del programador, y permite implementar la solución más natural para cada plataforma. Ésta es la causa de que a menudo C sea más eficiente que otros lenguajes. Típicamente, sólo la programación cuidadosa en lenguaje ensamblador produce un código más rápido, pues da control total sobre la máquina, aunque los avances en los compiladores de C y la complejidad creciente de los microprocesadores modernos han reducido gradualmente esta diferencia.

En algunos casos, una característica inexistente puede aproximarse. Por ejemplo, la implementación original de C++ consistía en un preprocesador que traducía código fuente C++ a C. La mayoría de las funciones orientadas a objetos incluyen un puntero especial, que normalmente recibe el nombre "this", que se refiere al objeto al que pertenece la función. Mediante el paso de este puntero como un

argumento de función, esta funcionalidad puede desempeñarse en C. Por ejemplo, en C++ se puede escribir:

```
stack.push(val);
```

Mientras que en C se podría escribir:

```
push(stack, val);
```

Donde el argumento stack es un puntero a una struct equivalente al puntero this de C++, que es un puntero a un objeto.

Tipos de variables en C.

Definición: Para la definición de variables se debe poner primero el tipo de variable a definir y luego el nombre de la variable (si queremos inicializarla podemos a continuación poniendo = valor) y si queremos definir mas de una variable del mismo tipo podemos poniendo una coma a continuación.

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
```

Tipo	Descripción	Bits	Rango de valores	Alias
SByte	Bytes con signo	8	[-128, 127]	sbyte
Byte	Bytes sin signo	8	[0, 255]	byte
Int16	Enteros cortos con signo	16	[-32.768, 32.767]	short
UInt16	Enteros cortos sin signo	16	[0, 65.535]	ushort
Int32	Enteros normales	32	[-2.147.483.648, 2.147.483.647]	int
UInt32	Enteros normales sin signo	32	[0, 4.294.967.295]	uint
Int64	Enteros largos	64	[-9.223.372.036.854.775.808, 9.223.372.036.854.775.807]	long
UInt64	Enteros largos sin signo	64	[0-18.446.744.073.709.551.615]	ulong
Single	Reales con 7 dígitos de precisión	32	[1,5×10 ⁻⁴⁵ - 3,4×10 ³⁸]	float
Double	Reales de 15-16 dígitos de precisión	64	[5,0×10 ⁻³²⁴ - 1,7×10 ³⁰⁸]	double
Decimal	Reales de 28-29 dígitos de precisión	128	[1,0×10 ⁻²⁸ - 7,9×10 ²⁸]	decimal
Boolean	Valores lógicos	32	true, false	bool
Char	Caracteres Unicode	16	['\u0000', '\uFFFF']	char
String	Cadenas de caracteres	Variable	Es permitido por la memoria	string

Fuentes:

- Rena, F. Manual de Programación en C para principiantes y avanzados
- [https://es.wikipedia.org/wiki/C_\(lenguaje_de_programación\)](https://es.wikipedia.org/wiki/C_(lenguaje_de_programación))
- <http://www.dc.fi.udc.es/~so-grado/current/Varios/CursoC.pdf>
- <http://www.monografias.com/trabajos4/lenguajec/lenguajec.shtml>
- http://www.ecured.cu/index.php/Lenguaje_de_Programación_C