Lenguaje de programación C

Orígenes

El lenguaje C fue inventado e implementado por primera vez por Dennis Ritchie. Nació en los Laboratorio Bell de AT&T, inspirado en el lenguaje B (basado en BCPL), escrito por Ken Thompson en 1970, a su vez, con intención de recodificar el sistema operativo UNIX , que en la fase de arranque está escrito en assembler, en vistas a su transportabilidad a otras máquinas B era un lenguaje evolucionado e independiente de la máquina. En 1972 , Dennis Ritchie, modifica el lenguaje B , creando el lenguaje C y reescribiendo el UNIX en dicho lenguaje.

Con la popularidad de las microcomputadoras se crearon muchas implementaciones de C. Como no existía ningún estándar, aparecieron discrepancias, Es así, que el instituto de estándares americanos (ANSI) estableció un comité para crear un estándar que definiera de una vez por todas el lenguaje C.

El lenguaje C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.

Características del lenguaje C

- Abundancia en operadores y tipos de datos.
- Codificación en alto y bajo nivel simultáneamente.
- Utilización natural de las funciones primitivas del sistema.
- Producción de código objeto altamente optimizado.
- Es estructurado.

El lenguaje C se programa con una serie de funciones que se llaman unas a otras para el procesamiento. Aun el cuerpo del programa es una función denominada flexible, permitiendo a los programadores la elección entre el uso de la biblioteca estándar que se provee con el compilador , el uso de funciones de terceros creadas por otros proveedores de C, o el desarrollo de sus propias funciones.

Una de las peculiaridades de C es su riqueza de operadores , puede decirse que prácticamente dispone de un operador para cada una de las posibles operaciones en código máquina.

A menudo se denomina al lenguaje C como un lenguaje de nivel medio, esto no significa que sea menos potente, más fácil de usar y menos evolucionado que los lenguajes de alto nivel. Se presenta como un lenguaje de nivel medio porque combina elementos de lenguajes de alto nivel con la funcionalidad del lenguaje ensamblador

El lenguaje C la unidad básica de programación es la función, los programas están formados por módulos. Cada módulo realiza una tarea específica y es un subprograma independiente.

El proceso de compilación

Cabe destacar que el único lenguaje que entiende la computadora es el lenguaje de máquina o sea ceros y unos por lo tanto hay que utilizar programas especializados que permitan convertir el lenguaje de alto nivel al lenguaje máquina.

Dichos programas se llaman compiladores e intérpretes y se refieren a la forma en que se ejecuta un programa.

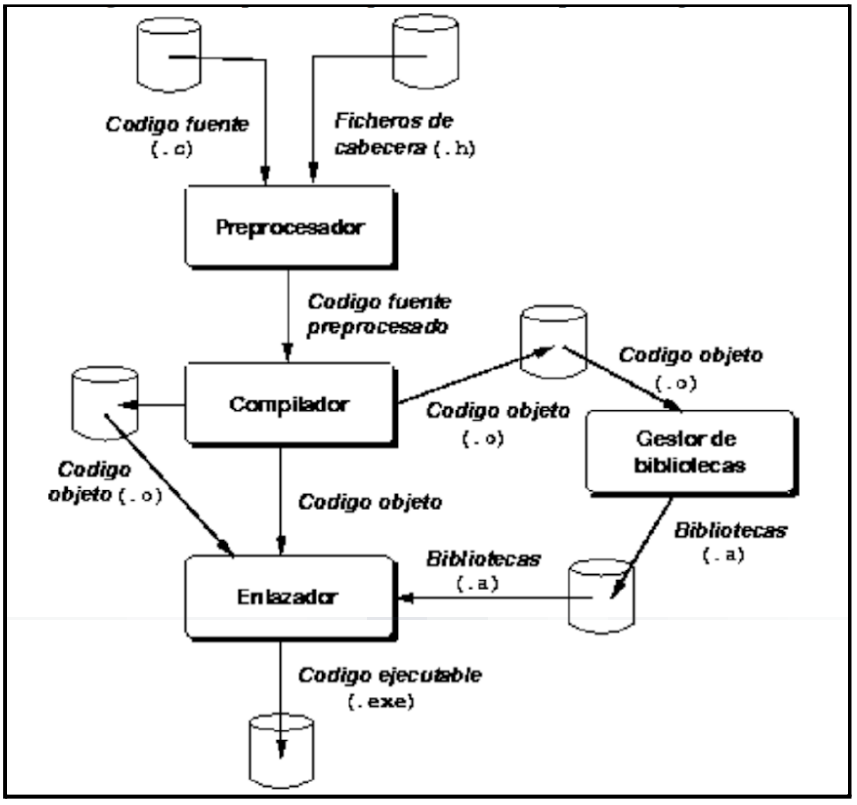
Compiladores

Un compilador analiza el programa y lo traduce al idioma "máquina". La acción fundamental de los compiladores es equivalente a la de un traductor humano , que toma nota de lo que está escuchando y reproduce por escrito en otra lengua.

Intérpretes

Analiza el programa fuente y lo ejecuta directamente, o sea en el ejemplo del traductor humano, éste sería un traductor humano que conforme a lo que está escuchando va ejecutando, sin generar ningún escrito, es decir que sobre la marcha va traduciendo.

El siguiente es un esquema básico del proceso de compilación de programas, módulos y creación de bibliotecas en C hasta llegar a obtener el código ejecutable.



El procesador

El preprocesador acepta como entrada código fuente y se encarga de:

- Eliminar los comentarios.
- Interpretar y procesar las directivas de preprocesamiento, precedidas siempre por el símbolo #.Dos de las directivas más comúnmente empleadas en C son #include y #define.

A lo largo del apunte profundizaremos en estas directivas y algunas otras más complejas.

Retomando el esquema mostrado en la Ilustración 1 el código fuente es traducido a lenguaje de máquina y el código resultante pasa directamente al compilador. Así, en la práctica el preprocesado se considera como la primera fase de la compilación.

El compilador

El compilador, analiza la sintaxis y la semántica del código fuente preprocesado y lo traduce, también tiene la función de reportar cualquier error en el programa fuente que detecte durante el proceso de traducción. Si tras la compilación se presentan errores en el programa fuente, es preciso volver a editar el programa, corregir los errores y compilar nuevamente, proceso que se repite hasta que no se producen más errores, generando así un fichero que contiene el código objeto

El enlazador

Si no existen errores en el programa fuente, se debe instruir al sistema operativo para que realice la fase de montaje o enlace (link), este proceso produce un programa ejecutable.

El enlazador resuelve las referencias a objetos externos que se encuentran en un fichero fuente, es decir a funciones de una biblioteca o de funciones que están definidas en otros archivos fuentes. El enlazador combina estas funciones con main() para crear un archivo ejecutable.

Estructura general de un programa en C

Un programa en C está compuesto de una o más funciones. La función principal main(), y una serie de directivas que permitirán incluir en el mismo archivos de cabecera.

La estructura básica se ve en la siguiente ilustración. Cabe destacar que en este ejemplo las líneas de programas están incompletas, simplemente se quería mostrar la forma general con el fin de identificar las distintas partes que lo conforman.

```c
/*Directivas de preprocesamiento, instrucciones que se dan al compilador
antes de ejecutar el programa principal*/
#include<libreria.h>
#define

//Declaraciones globales
 /*Prototipos de funciones
   Variables*/

//Función principal main
int main()
{
  /* Cuerpo del programa, incluye:
  - Declaración de variables "locales"
  - Sentencias
  - Llamadas a las funciones
  */
  return 0
}
```

Directivas de preprocesamiento

Los compiladores de C proporcionan bibliotecas de funciones. Cada biblioteca tiene asociada un archivo de definición que se denomina cabecera. Para utilizar algo de una biblioteca, hay que colocar al principio del programa una directiva de preprocesamiento seguida del nombre del archivo cabecera que por lo general tienen extensión h.

La directivas más usuales son: #include y #define.

Declaraciones globales.

Indican al compilador que todas las funciones y variables aquí declaradas son comunes a todo el programa y se sitúan antes de la función main.

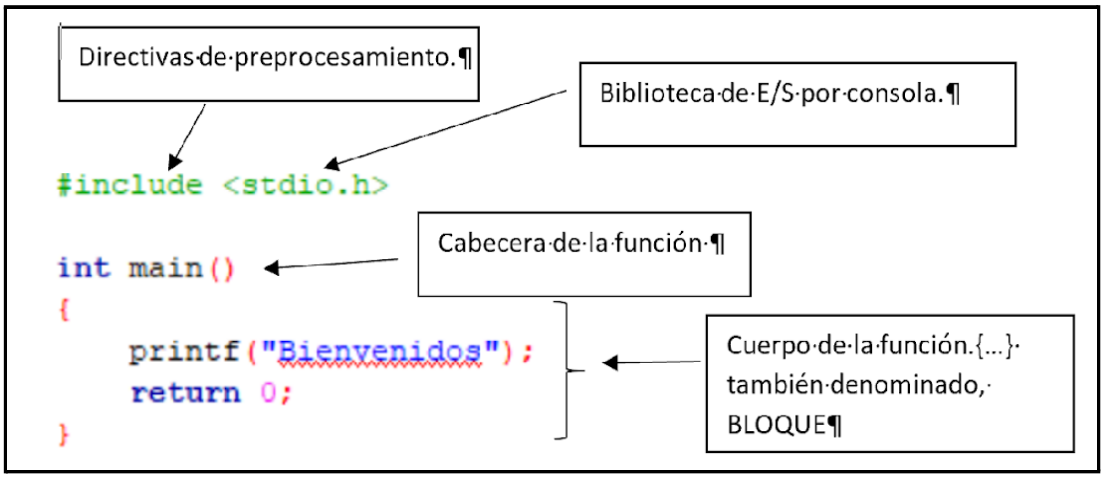
Si se declara una variable en esta zona podrá ser usada en todas las funciones que utilice el programa incluida la función principal. La declaración de funciones también se denomina prototipo (avanzaremos en esto cuando veamos el tema de funciones).

Función principal main()

Todos los programas tienen una cabecera que está indicada por la palabra main que es el punto de partida del programa propiamente dicho y solo debe haber un main por programa.

Dentro del bloque encerrado entre llaves {}, se encontrará el programa propiamente dicho junto con la declaración de las variables locales.

A continuación, la siguiente Ilustración es un ejemplo sencillo, a los fines de identificar las partes mencionadas en el párrafo anterior.



La directiva #include<stdio.h> incluye una serie de funciones de entrada/salida entre ellas, las dos más comunes son aquellas que transportan datos entre el programa y la entrada y salida estándar (teclado y pantalla respectivamente), scanf y printf.

El programa debe contener tantas directivas o #include como librerías sean necesarias incorporar, es decir deben especificar todos los ficheros de cabecera (ficheros con extensión .h) correspondientes a las librerías de funciones utilizadas. Iremos viendo más librerías en la medida en que avancemos en la programación o cuando el programa así lo solicite.

La sentencia return

La sentencia return tiene dos usos importantes. Primero, fuerza una salida inmediata de la función en que se encuentra. O sea, hace que la ejecución del programa vuelva al código que llamó a la función. En segundo lugar, se puede utilizar para devolver un valor.

Todas las funciones, excepto aquellas de tipo void, devuelven un valor. Este valor se especifica explícitamente en la sentencia return. Si no está la sentencia return, el valor devuelto por la función queda técnicamente indefinido.

Cabe destacar que main () a su vez es una función que devuelve un valor entero (int), por lo que necesitamos escribir la cláusula (return o), en c generalmente todo se trabaja a través de funciones y no siempre es necesario que devuelvan un valor, por lo tanto si la función no devuelve ningún valor se debe especificar utilizando la palabra reservada (void). Profundizaremos en esto cuando abordemos el tema de funciones.

Las Bibliotecas

El lenguaje C es un lenguaje muy reducido. Carece de tipos y funciones que forman parte de otros lenguajes. Es por esto por lo que, la mayoría de los programas incluyen llamadas a varias funciones contenidas en la biblioteca estándar de C.

Esta biblioteca define un conjunto de funciones, para llevar a cabo las tareas necesarias más comunes. Cuando se llama a una función que no es parte del programa que se ha escrito.

Por ejemplo, el lenguaje no incluye ninguna facilidad de entrada/salida, manipulación de cadenas de caracteres, funciones matemáticas, etc.

La biblioteca o librerías están clasificadas por el tipo de trabajo que hacen, hay librerías de entrada y salida, matemáticas, de manejo de textos, etc.

Ejemplos de funciones proporcionados por las bibliotecas estándares son:

- (stdio.h) entrada y salida de datos
- (string.h) manejo de cadenas
- (stdlib.h) memoria dinámica
- (math.h) rutinas matemáticas

Cabe destacar también que el programador puede desarrollar sus propias bibliotecas de funciones y enriquecer de esta manera su propio programa y por qué no el propio lenguaje.