



El lenguaje C

Origen

El lenguaje C fue creado hacia 1972 por **Dennis M. Ritchie** en un ordenador PDP-11. El primer objetivo que perseguía era crear un lenguaje sencillo que produjera programas de rápida ejecución, una especie de ensamblador avanzado, que permitiera crear estructuras complejas sin perder de vista las características del ordenador. Su origen se encuentra muy ligado al nacimiento del sistema operativo UNIX, creado, entre otros, por **Brian W. Kernigham**. UNIX está escrito en C, así como la inmensa mayoría de las aplicaciones UNIX. El libro clásico en el que se expone el C es *The C Programming Language*, de Kernighan y Ritchie, publicado en 1978.

En diciembre de 1989 se produjo una importante estandarización del C, creándose lo que se conoce como **ANSI C**, ahora universalmente reconocido.

Aplicación

El C se utiliza en una gran variedad de situaciones; se escriben en C sistemas operativos, aplicaciones científicas y comerciales, juegos, emuladores, compiladores de otros lenguajes, etc.

Es un lenguaje extremadamente flexible, que otorga poder absoluto al programador. Con él se crean programas muy rápidos. Todo esto tiene su contrapartida: cuando se comete un error en un programa, sus efectos pueden ser desastrosos.

Disponibilidad

Este lenguaje está disponible prácticamente para todos los ordenadores del mundo, sean grandes o pequeños, caros o baratos, empresariales o personales. Además, no resulta muy difícil escribir programas en C en un ordenador que se puedan compilar perfectamente en otro; así se dispone de programas **multiplataforma**.

Fases de la compilación

La compilación de programas en C está compuesta en realidad de varias partes bien diferenciadas. Es muy importante conocer estas fases para entender bien el lenguaje y para poder corregir los errores, que pueden aparecer en cualquiera de ellas. Cuando se invoca el compilador, es él quien va ejecutando las diferentes fases, por lo que, en principio, el programador sólo debe conocer el proceso, no dirigirlo.

En esquema, éste es el proceso que se sigue para crear un programa en C:

1. El programador escribe uno o más ficheros fuente, con extensión **c**, y, opcionalmente, uno o más ficheros de cabecera, con extensión **h** (del inglés *header*). En los ficheros de cabecera suele haber definiciones generales y en los fuente, el código.
2. El precompilador examina el código fuente de cada archivo **c**, realiza sustituciones de símbolos e incluye en el código el contenido de los ficheros **h**.
3. El compilador recoge la salida del precompilador y la convierte en código máquina, dejando el resultado en ficheros llamados **ficheros objeto**, que pueden tener extensión **o** o bien **obj**.
4. El montador (llamado en inglés *linker*) une los archivos objeto que provienen de los archivos del programador con otros archivos objeto imprescindibles y con las partes necesarias de los archivos de biblioteca (de extensión **a** o bien **lib**) y forma el archivo ejecutable, que en Microsoft Windows tiene extensión **exe** y en GNU/Linux no tiene extensión, sino permiso de ejecución.

Y gráficamente se puede ver así:



Entornos integrados de desarrollo

El proceso de creación de un programa, en cualquier lenguaje, exige repetir bastantes veces el proceso de compilación, para corregir los errores sintácticos que van apareciendo y repetir la ejecución del programa para corregir los errores lógicos. Se llama entorno de desarrollo a un programa que

permite editar el código fuente, lanzar el compilador, recoger los errores que éste genere y ejecutar el programa de forma dirigida para poder controlarlo. Estos entornos se denominan en inglés *Integrated Development Environment*, y se conocen con sus siglas: IDE.

Casi todos los compiladores ofrecen dos modos de trabajo: mediante un conjunto de programas independientes o mediante un IDE. Suele ser cuestión de gusto usar uno u otro modo.

Compiladores

Existe una gran cantidad de ellos, tanto libres como en propiedad. Destacaremos algunos:

Para GNU/Linux

En el mundo del software libre tiene una enorme importancia el compilador de C auspiciado por GNU, el `gcc` y el de C++, `gpp`. De hecho, es prácticamente el único que se utiliza en GNU/Linux. El diseño de este compilador permite que sea utilizado fácilmente en muchos otros sistemas operativos. El núcleo Linux se compila con `gcc`. Su mera existencia fue fundamental para que Linus Torvalds comenzara a escribir Linux.

Es el que se utilizará en todos los ejemplos de este curso, aunque éstos se podrán usar con cualquier otro compilador.



Para MS-DOS

- ◆ **DJGPP**. DJ Delorie llevó el `gcc` y el `gpp` a MS-DOS, creando el compilador DJGPP. Se puede obtener en <http://www.delorie.com>. Es un compilador extremadamente potente, con buena documentación, y una gran cantidad de herramientas traídas del mundo UNIX. Una gran cantidad de programas importantes (sobre todo libres) para MS-DOS se compilan con él.
- ◆ **Turbo C++**. Aunque fue comercializado hace muchos años por la empresa Borland, la versión 1.0 se ofreció gratuitamente a los lectores de la revista PC Magazine. Se puede descargar gratuitamente del “museo” de la empresa. Consúltese <http://www.borland.com>. Permite compilar programas en C y en C++, incluye muchas herramientas y un IDE; ocupa muy poco espacio en disco duro y funciona ágilmente en ordenadores de poca potencia.

Para Microsoft Windows

- ◆ **MinGW**. Este críptico nombre esconde el compilador “Minimalist GNU for Win32”, es decir, la versión de `gcc` para Windows. Por tanto, es software libre. Si no se dispone de una instalación GNU/Linux para seguir estas hojas, sino una instalación Windows, éste es el compilador recomendado. Su web es <http://www.mingw.org>.
- ◆ **C++ Builder Compiler**. El compilador más avanzado de Borland. Aunque es un programa comercial, se puede descargar gratuitamente de la página web de la compañía la versión más sencilla, que sólo incluye el compilador, sin IDE.
- ◆ **Microsoft Visual C++**. El producto más usado por los desarrolladores profesionales en la plataforma Microsoft Windows, ya que es un producto de Microsoft. Lleva licencia en propiedad.

GNU+Win32