C en GNU/Linux

Buena pareja

El lenguaje C se concibió, entre otras cosas, para realizar el sistema operativo UNIX; la mayoría de los programas del proyecto GNU se escriben en C; el núcleo Linux está escrito en C. Como se ve, el lenguaje C y el sistema operativo GNU/Linux se complementan perfectamente. Por eso, resulta muy adecuado desarrollar programas en C y aprender el lenguaje en este sistema operativo.

Editores

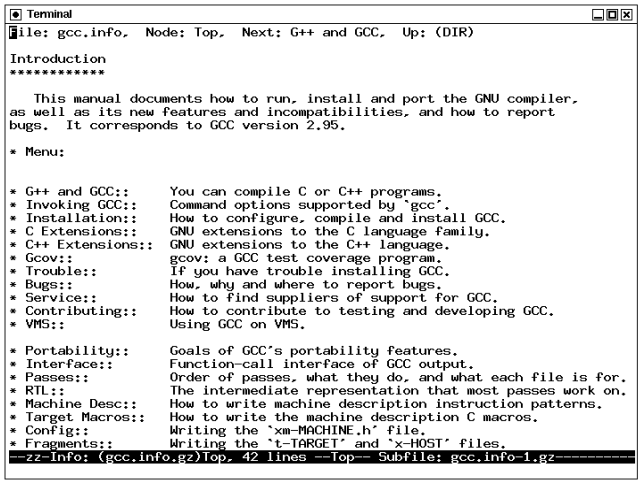
Para escribir el código es imprescindible un editor de textos, bien independiente o bien el editor del IDE. Ayuda mucho para escribir que el editor coloree de distinta manera cada parte del programa, lo que en inglés se llama *syntax highlighting*; muchos editores disponen de esta característica, como *emacs*, *kwrite* y *zed*.

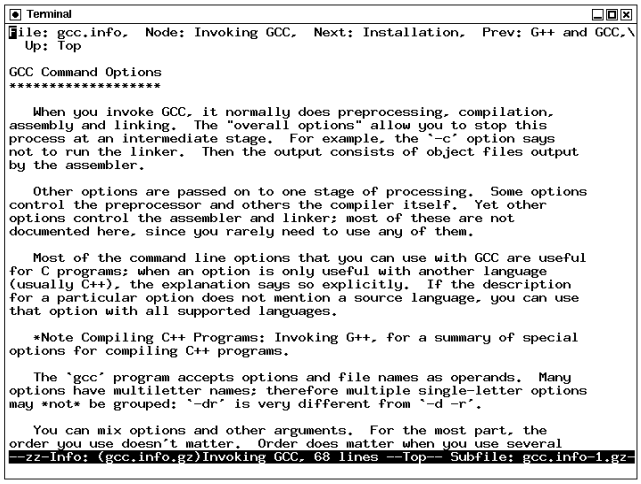
El compilador gcc

Es casi el único que se usa en GNU/Linux, ya que es el compilador del proyecto GNU. Es habitual que la instalación del sistema deje el compilador preparado para trabajar, pero si no es así, hay que instalarlo.

Documentación

La documentación de *gcc* se encuentra en formato info, por lo que se puede leer con varios programas diferentes: *info*, *GNOME Help Browser*, *Konqueror*. Véanse dos muestras:





Opciones básicas

Todos los compiladores de C admiten multitud de opciones, aunque para compilar programas sencillos no hacen falta más que unas pocas. Veamos un par de ejemplos, que serán suficientes para seguir el curso:

1. Para compilar los archivos **Fichero1.c** y **Fichero2.c** junto con la librería matemática y crear el ejecutable **Fichero**, se usa:

   `gcc -o fichero fichero1.c fichero2.c -lm`

   Como se ve, la opción **`-o`** sirve para indicar que el siguiente parámetro es el nombre que se desea dar al ejecutable; si no se usara, el ejecutable creado se llamaría **a.out**.
2. Para compilar los archivos **Fichero1.c** y **Fichero2.c** y crear únicamente sus archivos objeto se usa:

   `gcc -c fichero1.c fichero2.c`

   Queda de manifiesto que la opción **`-c`** es la que indica que sólo hay que realizar la compilación, pero no el montaje.
3. Para montar los archivos **Fichero1.o** y **Fichero2.o** y crear el ejecutable **Fichero**, se usa:

   `gcc -o fichero fichero1.o fichero2.o`

   Por tanto, es fácil separar las etapas de compilación y montaje, algo que para programas pequeños no es necesario, pero resulta imprescindible para programas grandes compuestos de muchos archivos.

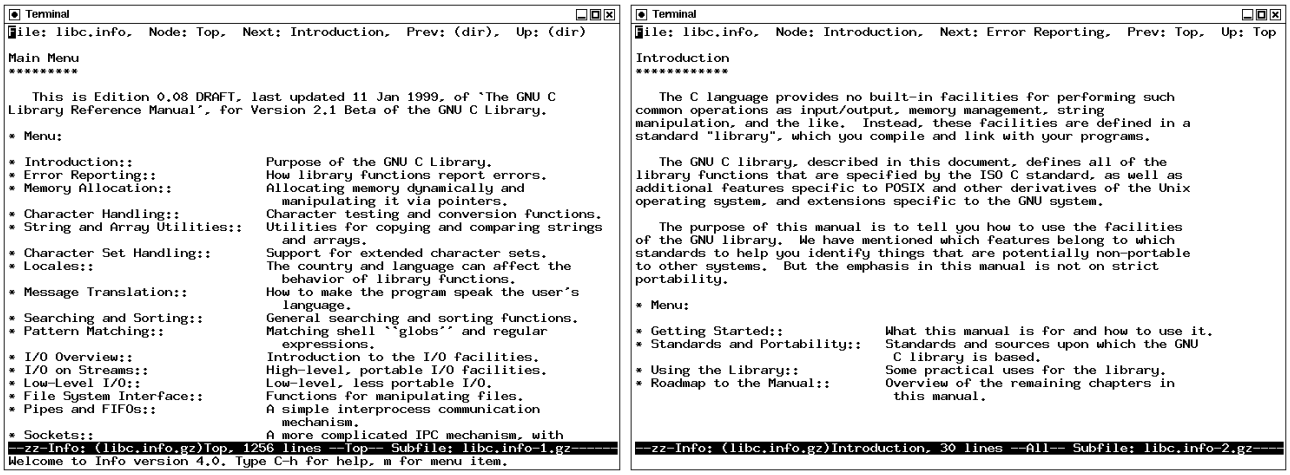
La librería glibc

El lenguaje C no incluye en sí mismo capacidades que en otros lenguajes se dan por supuestas, como imprimir datos en pantalla, por ejemplo. En vez de eso, existen funciones encargadas de realizar gran cantidad de tareas. La biblioteca que reúne las funciones fundamentales de C se llama libc, y glibc es la versión GNU de libc; glibc incorpora todas las características de libc y algunas más, propias y exclusivas.

En general, cuando se desea realizar alguna acción en la que intervenga el sistema operativo, hay que buscar la función correspondiente consultando la documentación de libc, que no es la misma que la de *gcc*. Una vez que se conocen las funciones que más se necesitan, la documentación sólo se consulta de vez en cuando.

Documentación

Se encuentra en formato info. Véase una muestra:



Programar para GNOME y KDE

La programación para entornos gráficos tiene algunas características que la hacen en principio diferente de la creación de programas para consola. Para crear un programa que tenga un interfaz GNOME o KDE, será necesario seguir ciertas reglas, utilizar algunas funciones específicas y montar el programa con librerías adicionales.

Entornos integrados

Existen en GNU/Linux varios entornos de integrados de programación. El más potente es *KDevelop*, que aunque funciona bajo KDE, permite crear programas para consola, para GNOME y para KDE. Un entorno menos potente es *Anjuta*, que funciona bajo GNOME.