



ÍNDICE

1. Introducción
2. Licencia General Pública GNU
3. Requisitos del sistema
4. Créditos
5. Primera ejecución
6. Pasos Básicos
 - 6.1. Creando un Proyecto
 - 6.2. Añadiendo y Quitando Archivos
 - 6.3. Opciones de Administración del Proyecto
 - 6.3.1. Opciones del Proyecto
 - 6.3.2. Linkando bibliotecas con tu Proyecto
7. Compilando y ejecutando
 - 7.1. Procesos de compilación y construcción
 - 7.2. Ejecutando tu programa
8. Depurando con Dev-C++
 - 8.1. Introducción a la Depuración de programas
 - 8.2. Depurando tu programa
 - 8.3. Comandos de Depuración integrados
 - 8.3.1. Puntos de ruptura
 - 8.3.2. Ejecución Paso a Paso
 - 8.3.3. Mostrando valores de variables y miembros de clases y estructuras
 - 8.3.4. Retroseguimiento
 - 8.3.5. Usando la ventana del CPU
9. Plantillas y paquetes de Dev-C++
 - 9.1. Plantillas integradas
 - 9.2. Consiguiendo e instalando paquetes
 - 9.3. Creando tus propios paquetes
10. Opciones
 - 10.1. Opciones del Compilador
 - 10.2. Opciones del Entorno
 - 10.3. Opciones del Editor
 - 10.3.1. Configuración del Editor
 - 10.3.2. Explorador de Clases / Completar Código
 - 10.3.3. Accesos rápidos (Atajos Teclado)
11. Obteniendo ayuda
 - 11.1. Preguntas Más Frecuentes FAQ
 - 11.2. Listas de Correo / Foro
 - 11.3. Contactando con los autores

1. Introducción

Dev-C++ es un entorno desarrollo integrado (IDE) para los lenguajes de programación C/C++. Usa Mingw basado en GCC (GNU Compiler Collection) como compilador. Puede crear ejecutables para Windows, ya sea en modo consola o GUI gráfico, así como también DLLs y bibliotecas estáticas. Dev-C++ puede también ser usado en combinación con Cygwin o cualquier otro compilador basado en GCC.

Dev-C++ es Software Libre distribuido bajo los términos de la Licencia General Pública GNU (GPL)

Las características de Dev-C++ son:

- Soporta compiladores basados en GCC (incluido Mingw)
- Depuración integrada (con GDB)
- Soporta múltiples lenguajes (idiomas)
- Explorador de Clases
- Explorador de Depuración de variables
- Autocompletar código
- Mostrado de funciones
- Administrador / Explorador de proyectos
- Editor con Sintaxis remarcada configurable
- Rápidamente crea programas en modo ventana, modo consola, bibliotecas estáticas y DLLs
- Soporta plantillas para crear tipos de proyectos propios personalizados
- Construcción con el Archivos de Construcción (Makefile)
- Edita y compila Archivos de Recursos
- Administrador de Herramientas
- Permite imprimir los documentos
- Utilidades de Búsqueda y Reemplazamiento
- Administrador de Paquetes, para una instalación de plugins y bibliotecas más fácil

2. Licencia

Bloodshed Dev-C++ se distribuye bajo los términos de la Licencia General Pública GNU. Asegúrate de leerla antes de usar Dev-C++. Consultar las condiciones.

3. Requisitos del sistema

Estos son los mínimos requisitos del sistema que utiliza Dev-C++:

- Microsoft Windows 95, 98, NT 4, 2000, XP
- 8 MB de RAM con un archivo de intercambio grande
- Procesador compatible Intel a 100 Mhz
- 30 MB de espacio libre en el disco duro

Estos son los requisitos recomendados de Dev-C++:

- Microsoft Windows 2000, XP
- 32 MB de RAM
- Procesador compatible Intel a 400
- 200 MB de espacio libre en el disco duro

4. Créditos

Desarrolladores: Colin Laplace, Hongli Lai, Mike Berg, Yiannis Mandravellos

Compilador Mingw: Mumit Khan, J.J. Var Der Heidjen, Colin Hendrix y desarrolladores GNU

Sistema de actualización, detección de errores y trabajo inicial en el archivo de ayuda en inglés: Kip Warner

Archivo de ayuda y traducciones a español: Fernando Muñoz Ledesma

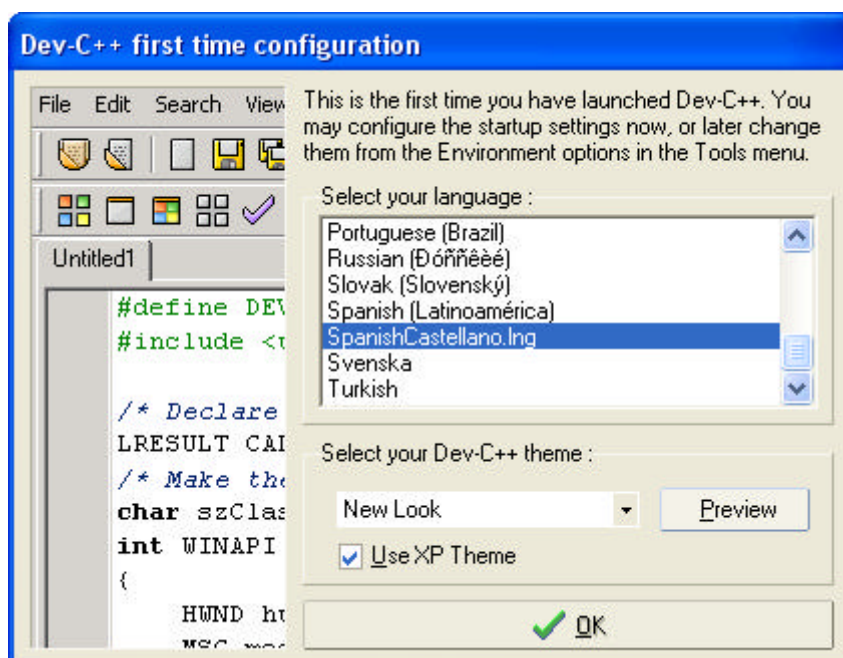
Tema New Look (Nuevo Estilo): Gerard Caulfield.

Iconos Gnome: Diseñadores Gnome

Tema Blue (Azul): Thomas Thron

5. Primera ejecución

Cuando inicias por primera vez Dev-C++, aparece el siguiente diálogo:



Puedes seleccionar el idioma que Dev-C++ usará, y también puedes escoger el tema de los iconos. También puedes activar el recuadro 'Use XP Theme' para que el entorno gráfico se muestre con el estilo de Windows XP.

Si has instalado una versión de Dev-C++ que incluye el compilador Mingw, todo lo que necesitas hacer es configurar el software, ya que los directorios del compilador vienen por defecto unos predefinidos. En este caso puedes seguir **6.Pasos Básicos**.

Sin embargo, si prefieres usar tu propia distribución de GCC (como Cygwin), puedes configurar manualmente los directorios del compilador y sus configuraciones para que se ajusten a tu sistema. Puedes hacer esto en **10.1. Opciones del Compilador**.

6. Pasos básicos

6.1. Creando un Proyecto

¿ Qué es un proyecto de Dev-C++ ?

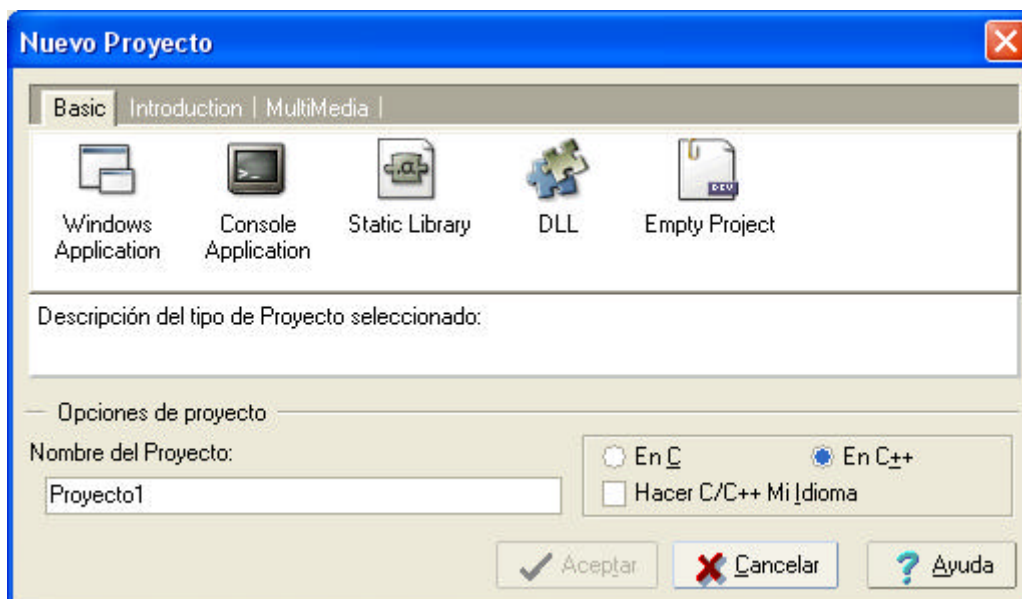
Un proyecto es un núcleo para administrar tus diferentes códigos y opciones dentro de Dev-C++. Te ayuda para visualizar tus distintos códigos fuente incluidos en el proyecto, y puedes fácilmente establecer distintos parámetros, como por ejemplo el tipo de programa que estás realizando (GUI, consola, DLL...).

¿ Cuándo usar Proyectos de Dev-C++ ?

- Si tienes más de un archivo con código fuente, debes crear un proyecto con el fin de que Dev-C++ una y construya tus archivos con código fuente juntos, después de compilarlos.
- Si necesitas crear un DLL o una biblioteca estática, o quieres usar Archivos de Recursos en tu programa.

¿ Cómo puedo crear un Proyecto de Dev-C++ ?

Ve al menú *Archivo* y haz clic en *Nuevo*, luego en *Proyecto*. Un diálogo que contiene los distintos tipos de Proyectos se abrirá.



Aquí se nombran los proyectos básicos:

- Aplicación de Windows (Windows Application): Crea un programa para Windows, que utiliza el API de Win32.
 - Aplicación en modo de consola (Console Application): Crea un programa en modo consola.
 - Biblioteca estática (Static Lib): Crea un proyecto vacío con las opciones necesarias para construir una biblioteca estática.
 - DLL (Dynamic Link Library DLL): Crea una Biblioteca de Enlace Dinámico DLL para Win32.
- Ahora cuando has seleccionado un tipo de proyecto, debes introducir el nombre de tu proyecto en donde corresponde y selecciona el lenguaje de programación que usarás en el proyecto C o C++, después haz clic en Aceptar.

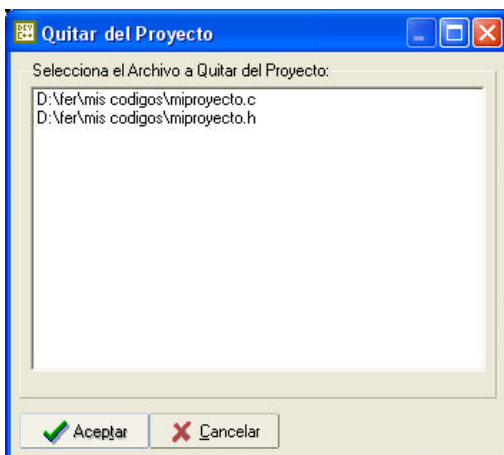
6.2. Añadiendo y Quitando Archivos

Añadir y eliminar archivos es muy fácil. Puedes añadir múltiples archivos al mismo tiempo haciendo clic en el menú *Proyecto*, y después en Añadir a Proyecto.

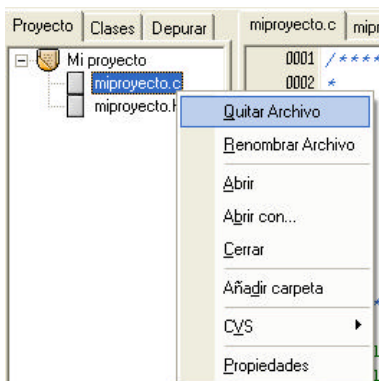


Tienes distintas formas para eliminar archivos de tu proyecto. Puedes hacer clic en el menú *Proyecto*, después en *Quitar del Proyecto* y seleccionar el archivo a quitar de la lista, o puedes hacer clic en el botón derecho en el Explorador de Proyectos y hacer clic en *Quitar*.

Pulsando en el menú Proyecto, después Quitar . . .



Haciendo clic con el botón derecho en el Explorador de Proyectos . . .



6.3. Opciones de administración del proyecto

6.3.1. Opciones del proyecto

Puedes cargar las Opciones del Proyecto haciendo clic en el menú *Proyecto* y después en *Opciones* (o presionando las teclas Alt + P).

Pestaña de Principal:

Nombre: Modifica aquí el nombre de tu proyecto.

Icono: Puedes asignar un icono a tu programa, ya sea seleccionando uno en la Biblioteca de Iconos, o añadiendo tu propio icono usando el botón *Examinar*.

Tipo: Esta es una configuración importante que indica el tipo de proyecto que estás haciendo.

Selecciona:

- **Win32 GUI:** si quieres que tu aplicación sea un programa gráfico para Windows.
- **Win32 Console:** si quieres que tu aplicación necesite una ventana de consola.
- **Win32 Static Lib:** si estás creando una biblioteca estática.
- **Win32 DLL:** si estás creando una biblioteca de enlace dinámico (DLL).

Pestaña de Archivos :

Esta ventana te permite modificar los parámetros de compilación y las opciones para cada código fuente.

Prioridad de construcción: Incrementa este valor para que el código tenga más prioridad al compilar que los demás.

Incluir en la compilación: Si está desactivado, tu código fuente no se compilará.

Incluir en link: Añade el archivo objeto generado del código fuente a la hora de linkar el proyecto.

Compila como C++: Activa esto si quieres que tus archivos se compilen como si fuesen códigos fuente de C++.

Anteponer a la construcción: Para usuarios expertos sólo. Aquí puedes cambiar los parámetros usados por Dev-C++ para compilar tu archivo.

Parámetros del compilador: Usa esto para especificar parámetros que pasar al compilador

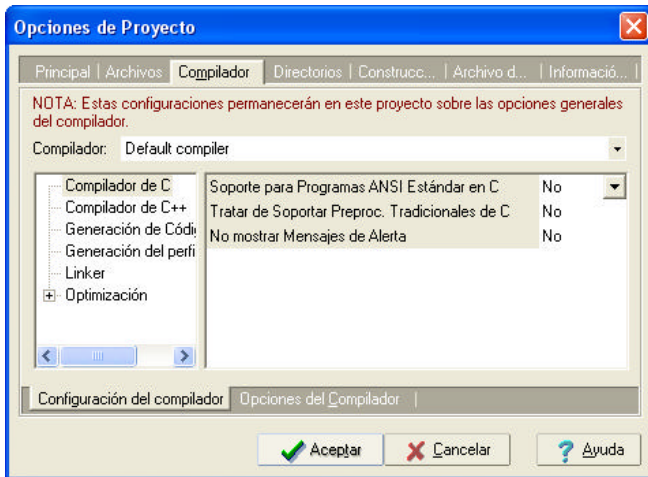
Parámetros del Linker: Usa esto para especificar los parámetros a pasar al linker, o para linkar bibliotecas con tu proyecto (6.5.).

Pestaña del compilador :

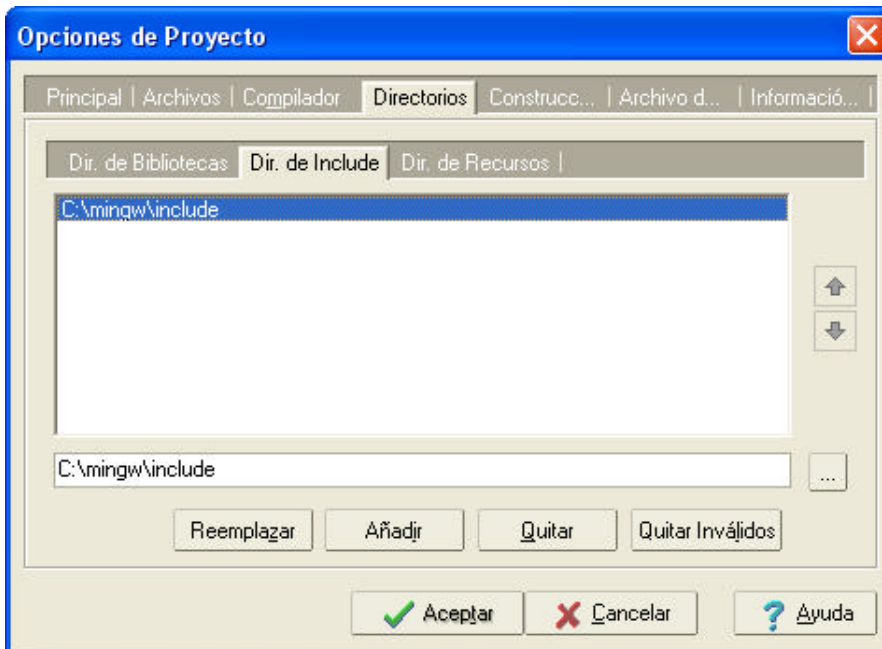
Mira **10.1. Opciones del compilador.**

Pestaña de parámetros :

Puedes especificar aquí los argumentos que se pasarán al compilador de C/C++ y al linker. Utiliza los parámetros del linker para especificar bibliotecas que linkar con tu proyecto. Para más información, lee linkar bibliotecas con tu proyecto (6.5.).



Pestaña de Directorios:



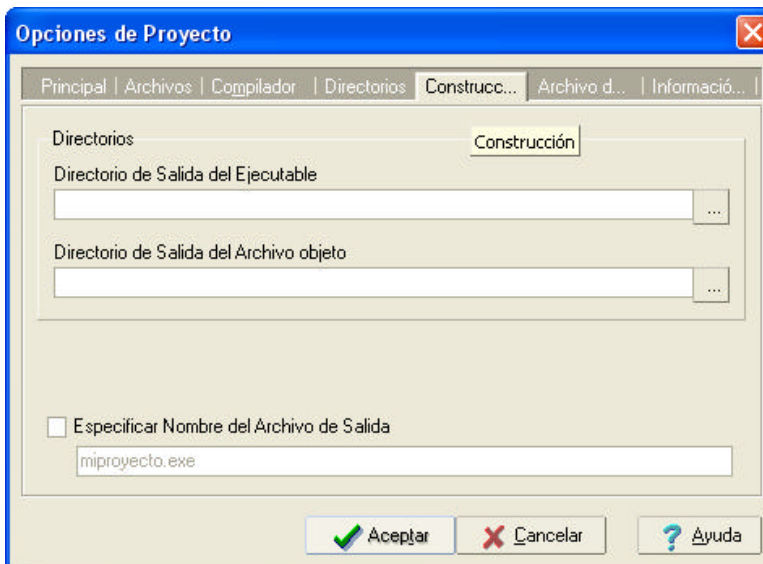
Puedes incluir aquí una lista de Includes, Archivos de Recursos y Directorios de Bibliotecas donde se buscarán archivos a la hora de la compilación y link.

Usa la pantalla de *Archivos de Recursos* para especificar tus *Archivos de Recursos* (Archivos *.rc).

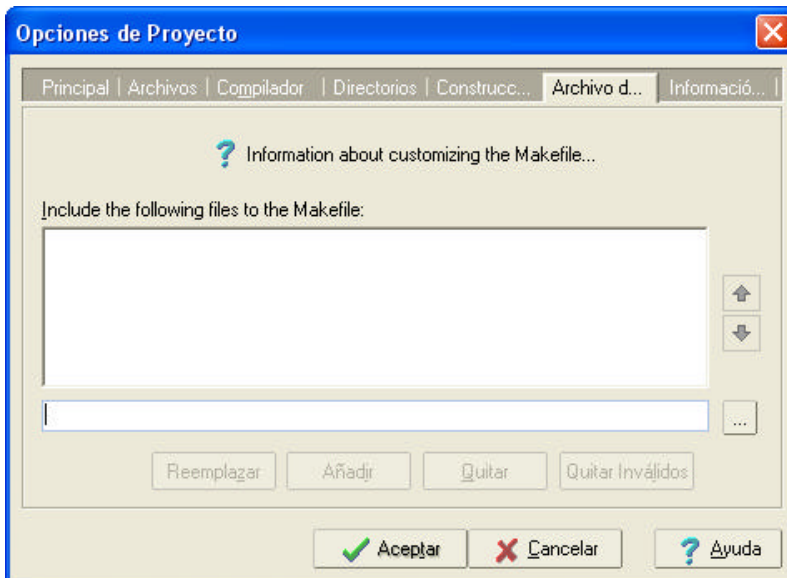
Pestaña de Construcción:

Directorio del ejecutable de salida: Puedes especificar un directorio para tu archivo ejecutable. El archivo ejecutable se creará en dicho directorio (si no se especifica ninguno, se crea en el directorio del proyecto).

Directorio del Archivo Objeto: Puedes especificar aquí el directorio donde los Archivos Objeto se crearán (si no se especifica ninguno, se crea en el directorio de los códigos fuente). Especificar el nombre del archivo de salida: Quizá quieras cambiar el nombre del ejecutable de salida aquí.



Pestaña del Archivo de Construcción:



Dev-C++ crea automáticamente un Archivo de Construcción para el proceso de Construcción. Si tienes experiencia con Archivos de Construcción y quieres añadirle líneas, aquí puedes hacerlo.

Pestaña de Información de la versión:

Aquí puedes especificar información adicional sobre la versión de tu programa.

6.3.2. Linkando bibliotecas con tu proyecto

Un poco de historia:

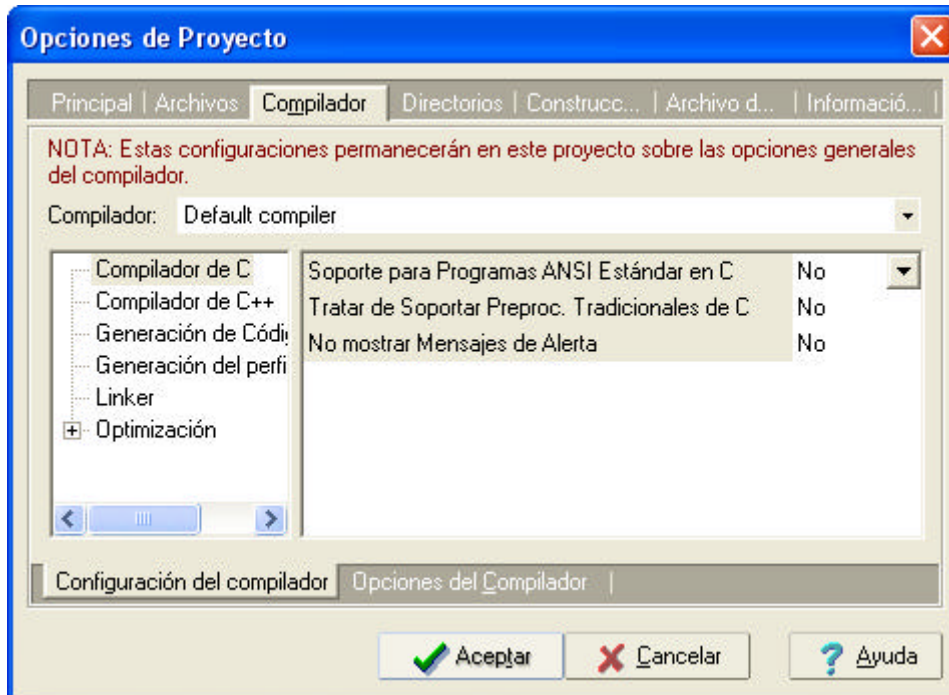
Los nombres de los archivos de Bibliotecas en un sistema GNU eran de la forma *libNOMBRE.a* (siendo NOMBRE el nombre de la biblioteca, como por ejemplo *wsock32*).

Si por ejemplo necesitabas usar la biblioteca *wsock32* (*winsock*), el nombre del archivo sería *libwsock32.a*

El parámetro que debía pasarle a GCC para linkarla es `-lnombre`, entonces para linkar la biblioteca pasaríamos el parámetro `-lwsck32` a GCC.


Linkando tu biblioteca:

Haz clic en el menú *Proyecto* y luego en *Opciones*. Ve a la pantalla del *Compilador*, y luego a *Opciones del Compilador*.



En la caja editable del *Linker*, puedes especificar tantas bibliotecas como necesites. Puedes también pasar el nombre completo de la biblioteca.

Ejemplo : `-lwx -lm -lwsck32 c:\bibliotecas\mibiblioteca.a c:\objetos\miobjeto.o`

También puedes utilizar el botón *Añadir Biblioteca u Objeto*  para seleccionar tu biblioteca de una lista.

7. Compilando y ejecutando


7.1. Procesos de Compilación y Linkado

¿ Cómo funcionan o trabajan los procesos de compilación y linkado ?

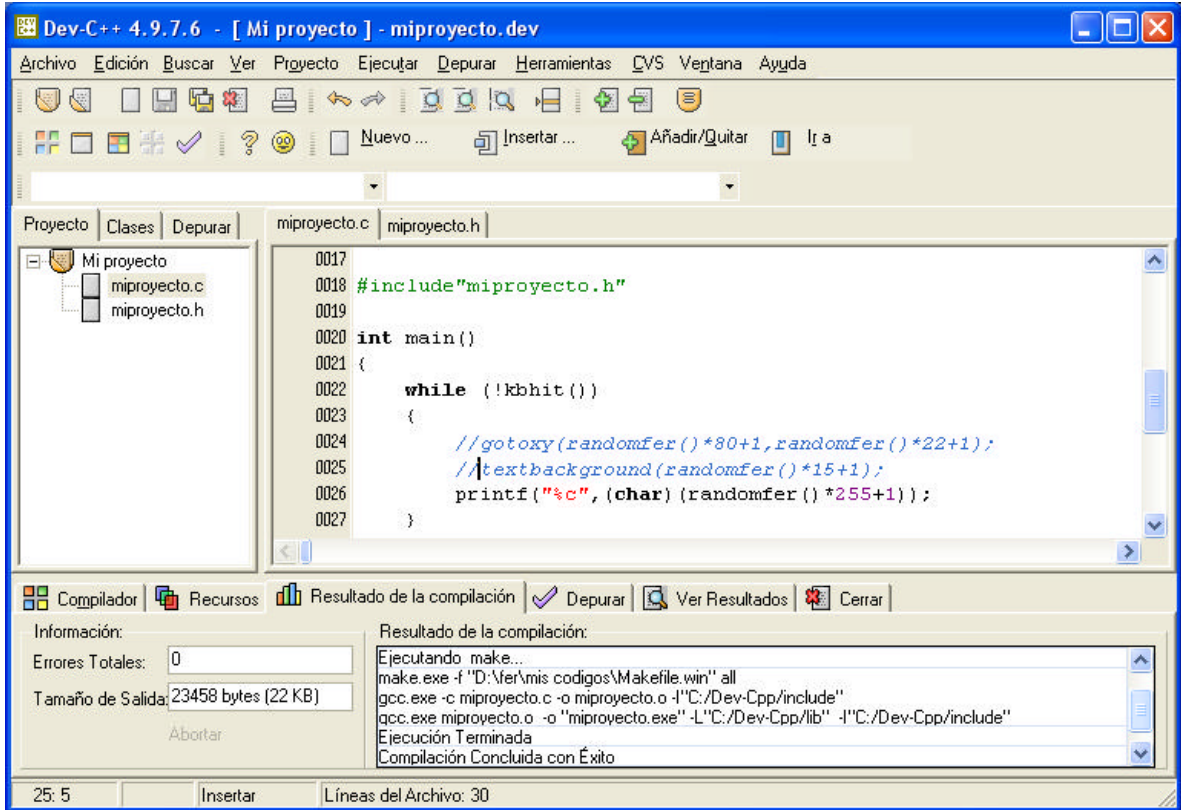
El proceso se puede dividir en diferentes programas:

- **Preprocesador** : procesa el código fuente antes de compilarlo. Por ejemplo, chequea y reemplaza Macros y Archivos Incluidos (`#include`) en tu código.
- **Compilador** : transforma tus archivos de código fuente en código ensamblador (lenguaje del procesador que las personas puedan entender)
- **Ensamblador** : coge el código ensamblador y genera código objeto binario, que sólo entiende la máquina.
- **Linker**: ensambla y une objetos juntos para crear un único ejecutable.

¿ Cómo hacer esto en Dev-C++ ?

Simplymente ve al menú *Ejecutar* y haz clic en *Compilar*  (o presionando : Ctrl+F9), y Dev-C++ será el que se preocupe de llamar al compilador y al linker.

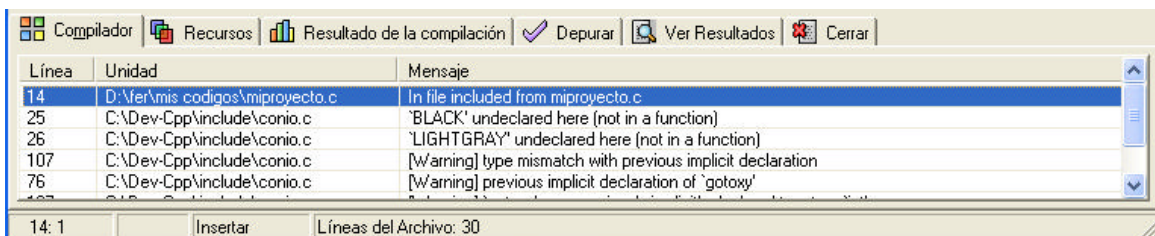
Mira el panel de abajo de Dev-C++, y aparecerá algo parecido a esto:



Esta ventana de información de la compilación muestra que está haciendo Dev-C++ (los programas que ejecuta, los archivos que está creando,...).

En el caso de tu programa se compile, verás el mensaje 'Compilación Concluida con Éxito' en esta ventana.

Si el proceso de compilación o linkado fallan, verás una lista de errores en dicho panel:




Puedes saltar a la línea incorrecta de tu código directamente haciendo doble clic donde aparece la línea con la descripción del error.

Nota: 'parse error' es un error de sintaxis (puede faltar un punto y coma al final de la línea, etc...).

7.2. Ejecutando tu programa

Fundamentos :

Ejecutar tu programa es tan sencillo como hacer clic en el menú *Ejecutar*, luego en *Ejecutar*  (o presionando: Ctrl+F10).

Puedes también usar *Compilar y Ejecutar*  (o presionando F9) para construir tu programa antes de ejecutarlo.

Si necesitas pasarle parámetros a tu programa, puedes usar el botón *Parámetros* dentro del menú *Ejecutar* y especificarle los parámetros.

Perfiles :

Este proceso dice que sucede en tu programa (llamadas a funciones, tiempo de ejecución,...) cuando lo ejecutas.

Tienes que activar la opción *Generar información del perfil para análisis* en *Opciones del Compilador*, en *Generación del perfil del Código* en *Configuración*. Luego compila tu proyecto como generalmente se hace, y ejecútalo (la información del perfil se generará y será refrescada cada vez que ejecutes el programa).

Cuando esté listo, haz clic en el menú *Ejecutar*, después en *Análisis del perfil*. La ventana del perfil aparecerá y mostrará el perfil de ejecución de tu programa.

8. Depurando con Dev-C++


8.1. Introducción al proceso de depurado

Es un hecho conocido que algunas veces desarrolladores cometen errores cuando hacen un programa. Estos errores reciben el nombre de 'bugs' o 'pulgas' y pueden ser difíciles de encontrar y corregir, sin embargo el depurador se ha creado para ayudar a los desarrolladores a investigar su programa cuando está ejecutándose.

Un depurador básicamente ejecuta un programa mientras se queda con los pasos y pistas de sus funciones, variables e instrucciones. Es capaz de parar el programa en cualquier momento dado, en cualquier Punto de Ruptura. Puedes establecer Puntos de Ruptura en cualquier parte de tu código: una vez que tu programa alcance dicho código durante su ejecución, el depurador parará y te dejará examinar los datos actuales del programa.

Probablemente sepas ya lo útil que puede ser el depurador a la hora de intentar resolver errores, y por tanto querrás pasar a la siguiente sección para aprender cómo utilizar la depuración integrada de Dev-C++.

8.2. Depurando tu programa

Correr tu programa en el depurador es fácil. Simplemente ve al menú *Depurar* y haz clic en *Depurar*  (o presiona F8).

Si no tienes establecida la información para poder depurar en tu proyecto, Dev-C++ te pedirá si quieres reconstruir tu proyecto con dicha información activada. Puedes seleccionar

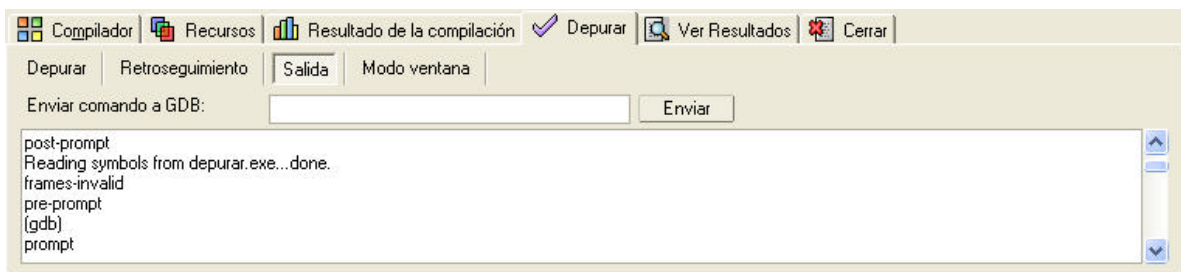
manualmente esa opción en *Opciones del Compilador* in la sección de *Configuración*. Después de que tu proyecto haya sido reconstruido, puedes ejecutar la depuración de nuevo.

El depurador ha cargado ahora tu programa y lo ejecuta, pero antes debes aprender a controlar varios aspectos del depurador:

- Estableciendo puntos de ruptura en tu código (8.3.1.).
- Ejecución paso a paso de tu código (8.3.2.).
- Mostrando el valor de las variables y miembros de clases y estructuras (8.3.3.).
- Retroseguimiento (8.3.4.).
- Usando la ventana del CPU (Procesador) (8.3.5.).

La salida que Dev-C++ recibe de GDB se muestra en la hoja de *Salida de Depuración* en la parte de debajo de Dev-C++.

Esta salida es interesante probablemente sólo si estás familiarizado con GDB. Puedes enviar también comandos directamente a GDB (si no conoces los comandos de GDB puedes enviarle el comando *help* para tener la lista) usando la caja editable justo encima de la salida (mira la pantalla ejemplo adjunta bajo estas líneas).

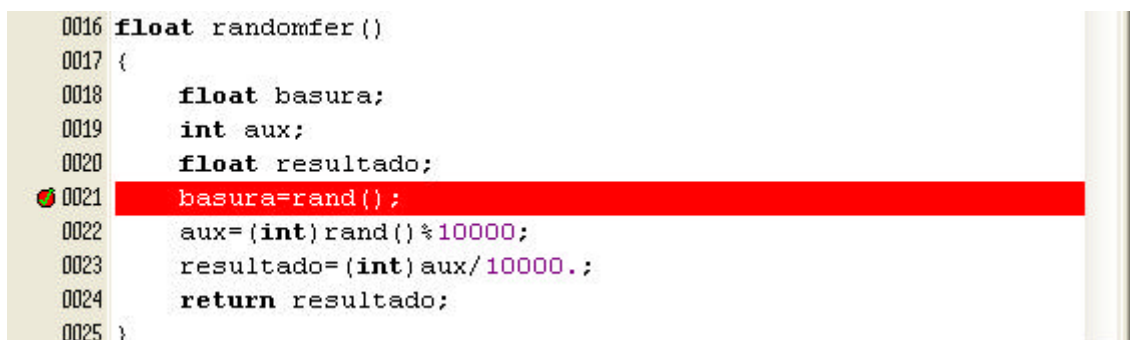


8.3. Comandos de Depuración integrados

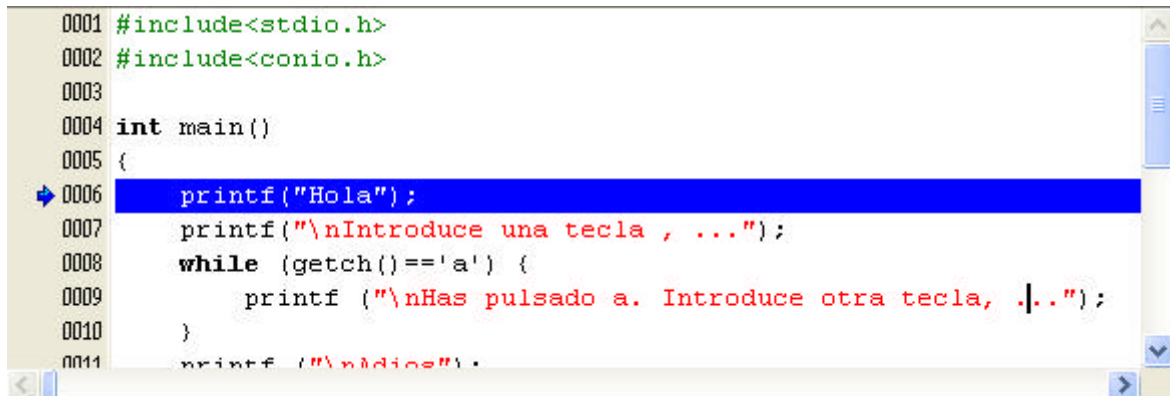
8.3.1. Estableciendo Puntos de Ruptura en tu código

Puedes usar Puntos de Ruptura para pausar tu programa en cierta instrucción (línea de código).

Para añadir un Punto de Ruptura, selecciona primero la línea de código donde quieres parar. Ahora haz clic en el menú *Depurar* y luego en *Añadir / Quitar Punto de Ruptura* (o pulsando Ctrl+F5). Haciendo clic en la barra en el margen izquierdo del editor a la altura de la línea tendrá el mismo efecto que los pasos anteriores:



Cuando cargue el depurador (**8.2. Depurando tu programa**), si tu programa ejecuta la línea de código con el Punto de Ruptura, Dev-C++ te avisará que se ha llegado a la línea cambiándola el color del fondo a azul:





```
0001 #include<stdio.h>
0002 #include<conio.h>
0003
0004 int main()
0005 {
0006 printf("Hola");
0007 printf("\nIntroduce una tecla , ...");
0008 while (getch()=='a') {
0009     printf ("\nHas pulsado a. Introduce otra tecla, .!..");
0010 }
0011 printf ("\n¡¡¡¡");
```

Ahora puedes examinar los datos en tu programa (8.3.3.) o ejecutar paso a paso tu programa (8.3.2.).

8.3.2. Ejecución paso a paso de tu programa

Una vez que se llegue a un Punto de Ruptura, puedes saltar pasos en tu programa de las distintas formas siguientes:

- **Siguiente Paso**  (o presionando F7) : El depurador dará un paso a la siguiente instrucción (línea de código) en tu programa.
- **Step Into** (o presionando Shift+F7) : El depurador saltará una instrucción. Si esa instrucción es una llamada a una función saltará a ella.
- **Continuar**  (o presionando Ctrl+F7) : El depurador continuará la ejecución del programa hasta que haya un Punto de Ruptura o el Final del código.

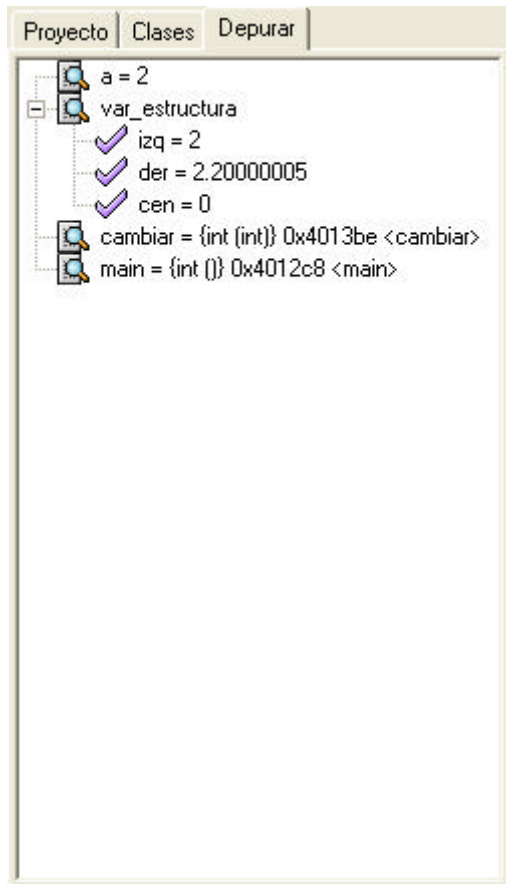
8.3.3. Mostrando valores de variables y miembros de clases y estructuras

Depurar es interesante también por la posibilidad de mostrar el valor de las variables observadas en un determinado instante.

Dev-C++ te permite ver en una conveniente manera, los contenidos de tus clases, cadenas, estructuras/uniones, arrays y otras variables, en el Explorador de Depuración de Variables, (figura que se muestra más abajo).

Puedes observar tus variables (cuando se llega a un punto de ruptura) de dos formas distintas:

- Haciendo clic en el botón *Añadir Watch* (o presionando F4), y escribiendo el nombre de la variable en el diálogo, y presionando Aceptar. Si seleccionas una palabra en el código fuente y presionas F4, se añadirá la variable observada sin preguntar por el nombre.
- Apuntando con el ratón sobre una variable en tu código fuente si la opción *Observar la variable seleccionada* (apuntada por el ratón) está activada en opciones del entorno (**10.2.**) y se añadirá a la lista de variables observadas.



Notas importantes:

- Cuando se usan punteros de estructuras o clases, si quieres ver todos los miembros de la variable *mi_puntero* entonces necesitas observar **mi_puntero* ('*' es el operador para el valor). Observando sólo *mi_puntero* mostrará exactamente la dirección de memoria contenida en *mi_puntero*.

- Algunas veces el depurador no sabrá de que tipo es un puntero, y puede no mostrar todos los miembros de la estructura o clase a la que apunta. Puedes solucionar este problema haciéndole un casting al puntero. Por ejemplo, si el depurador no puede mostrar los miembros de la variable apuntada por *mi_puntero* que es de tipo *mi_tipo* puedes probar añadiendo la variable observada: **(mi_tipo *)mi_puntero*.

8.3.4. Retroseguimiento

Retroseguimiento es un concepto de depuración que te dice que funciones se llamaron antes de llegar a un punto de ruptura o a una interrupción (como una violación de acceso). Para ello he aquí un ejemplo sencillo:

```
#include <stdio.h>
void test2()
{
    printf("hola\n"); /* Pon un Punto de ruptura en esta línea */
}

void test()
{
    test2();
}

int main(int argc, char **argv)
{
    test();
    return 0;
}
```

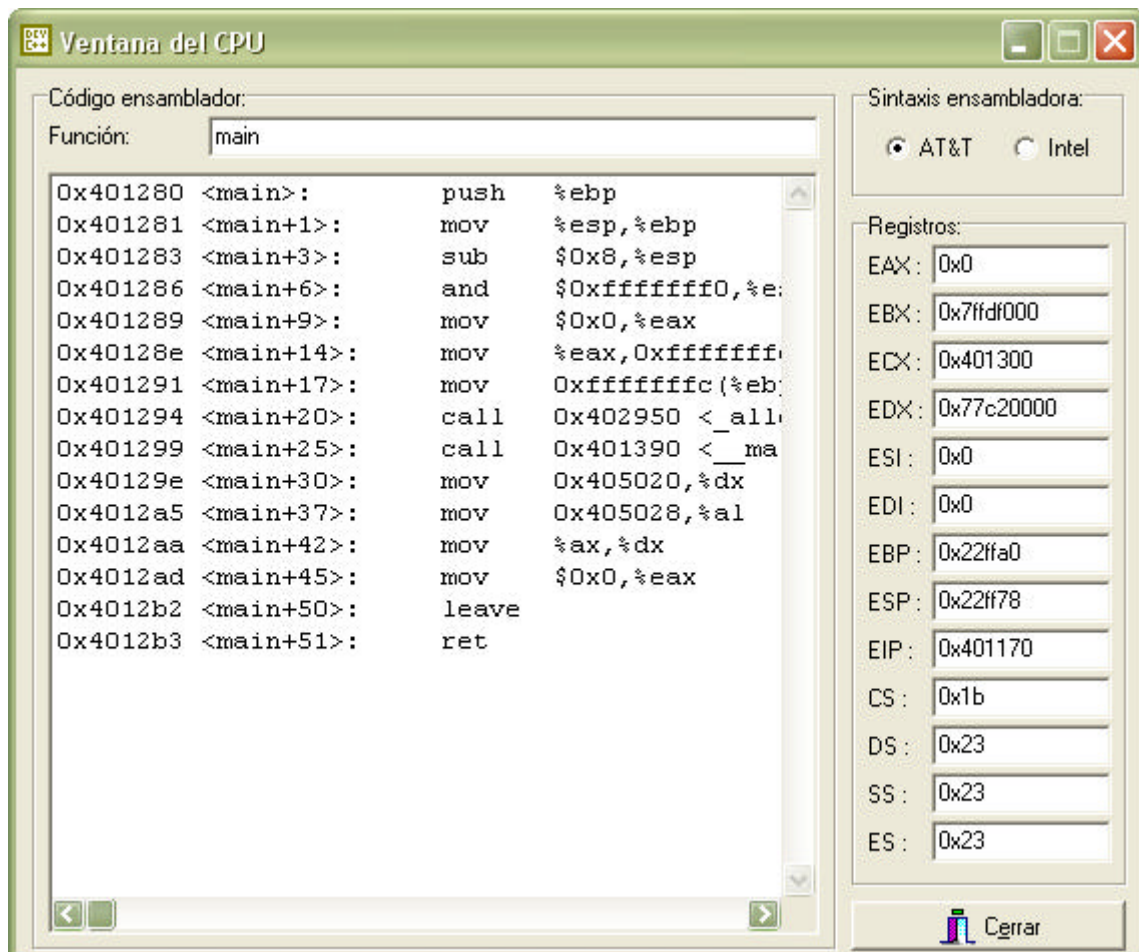
Pon un punto de ruptura en la línea del printf en la función test2() en el printf(), y carga el depurador. Al punto de ruptura se llegará casi instantáneamente. Ahora ve al diálogo de *Depurar* y haz clic en el botón de *Retroseguimiento*.

Esta muestra correctamente la lista de funciones que se han llamado (desde el comienzo del programa) antes de llegar al punto de ruptura. Haciendo clic en una función de la lista te mostrará su implementación en tu código fuente.

8.3.5. Usando la ventana del CPU

Dev-C++ está provisto de una ventana del CPU para desarrolladores expertos que quieran acceder al valor de los registros y a las instrucciones del CPU.

Para ver la ventana del CPU, espera a un punto de ruptura o una interrupción que pare tu programa y yendo al menú *Depurar* y seleccionando *Ventana del CPU* el siguiente diálogo aparecerá en pantalla:



A la izquierda puedes ver las instrucciones en ensamblador para la función actual. Puedes ver el código ensamblador de cualquier función escribiendo su nombre en el campo de *Función* y luego presionando *Enter*. Puedes seleccionar la sintaxis del código ensamblador: AT&T or Intel. A la derecha, se muestra el contenido de los registros del CPU.

9. Plantillas y paquetes de Dev-C++

9.1. Plantillas integradas

Dev-C++ utiliza un sistema de plantillas que es capaz de crear proyectos específicos rápidamente. Las plantillas son generalmente un conjunto de códigos fuente y un archivo de proyecto que incluye un código mínimo y todas las opciones del compilador y del linker, las cuales son necesarias para compilar ese tipo de programa.

10. Opciones

10.1. Opciones del Compilador

Para acceder al diálogo de Opciones del Compilador haz clic en el menú *Herramientas* y luego en *Opciones del Compilador*.

Compilador

Compilador a configurar

Aquí, puedes tener distintos perfiles de compiladores. Cada perfil tiene sus propias opciones y configuraciones, y así podrías cambiar entre distintos compiladores (por ejemplo, MinGW y Cygwin). El compilador por defecto que viene en el paquete de Dev-C++ es MinGW. Hay tres botones: Para añadir, eliminar y modificar un compilador.

Añadir los siguientes comandos al llamar al compilador

Aquí puedes especificar opciones adicionales de línea de comandos que serán pasadas al compilador GCC cuando se compile tu proyecto o código fuente. Revisa el manual de MinGW para una lista de comandos pasables como argumentos (Esta disponible en su sitio web: www.mingw.org).

Añade estos comandos a la línea de comandos del linker

Aquí puedes especificar opciones que se pasarán al linker (ld.exe). Puedes especificar bibliotecas aquí.

Retraso en la compilación

Esta opción está presente para añadir un retraso antes de compilar. Normalmente, no usarás esto. Si el proceso de construcción se queja de que la fecha del archivo es incorrecta, intenta poniendo un cierto retraso aquí.

Utiliza una generación de dependencias rápida pero imperfecta

Por defecto, Dev-C++ comprobará todos los archivos y cabeceras para la información de dependencias, y actualizar el archivo de construcción de tal forma. Si ves que está tomando mucho tiempo, puedes prevenirlo activando esta opción.

Configuraciones / Compilador de C

Soporta todos los programas ANSI C estándar

Opción -ansi: Intentará ser todo lo compatible con ANSI posible.

Intentar soportar algunos aspectos de preprocesadores de C tradicionales

Opción -traditional-cpp: Intentará hacer que el preprocesador se comporta como lo hacían los tradicionales.

Inhibe mensajes de advertencia

Opción -w: No se mostrarán mensajes de advertencia.

Configuraciones / Compilador de C++

Apagar todos los chequeos de accesos

Opción -fno-access-control

Aceptar \$ en identificadores

Opción -fdollar-in-identifiers: Permitirá que \$ sea usado en los nombres de las funciones y variables. No se puede usar en GCC 3.2.

Usar heurística para compilar más rápido

Opción -fsave-memoized

Configuraciones / Generación de Código

Activar manejo de excepciones

Opción -fexceptions

Usar el mismo tamaño para double que para flota

Opción -fshort-double

Poner información extra en el código ensamblador creado

Opción -fverbose-asm: El código ensamblador intermedio de los archivos tendrá información extra en forma de comentarios. Sólo es útil si son guardados (usando el comando en línea de comandos **-S**).

Configuraciones / Generación del perfil del código

Generar información del perfil para análisis

Opción -pg: escribe información extra en el programa generado para usarla en el análisis del perfil. El análisis del perfil te permite saber qué cantidad de tiempo de ejecución utiliza tu programa, y así cuando lo optimices intentar optimizar más las partes que marcan las diferencias mayores. Esta opción debería desactivarse en la versión final / retail. Debe usarse sólo para construcciones para depuración.

Configuraciones / Linker

Linkar un programa de C Objetivo

Opción `-lobjc`

Generar información para la depuración

Opción `-g3`: Escribe información para depurar en los binarios generados. Esto permite depurarlo con el depurador integrado.

No usar archivos o bibliotecas del sistema

Opción `-nostdlib`

No crear una ventana de consola

Opción `-mwindows`: Obliga a GCC a construir una aplicación con interfaz gráfico GUI para Windows.

Configuración / Optimización

Realizar un menor número de optimizaciones

Opción `-fexpensive-optimizations`

Configuración / Optimización / Optimizaciones intensivas

Aquí puedes seleccionar un nivel de optimización:

Optimizar

Opción `-O1`

Optimizar más

Opción `-O2`

Mejor optimización

Opción `-O3`

Directorios

Binarios

Especifica la localización del compilador (ejecutables).

Bibliotecas

Especifica la localización de los archivos de bibliotecas (*.a).

C includes

Especifica la localización de los archivos de cabecera para los programas en C (*.h).

C++ includes

Especifica las localizaciones de los archivos de cabecera para los programas de C++ (*.h, *.hpp).

Programas

Aquí puedes especificar los nombres de los archivos de los componentes ejecutables de los compiladores.

10.2 Opciones del entorno

Para acceder al diálogo de Opciones del Entorno, haz clic en el menú 'Herramientas' y luego en 'Opciones del Entorno'.

Principal

C++ por defecto en nuevo proyecto

Cuando haces un nuevo proyecto, se asume que esté en C++. Quitando esta opción asumirá que usas C.

Crear Archivos de Respaldo (Copias de Seguridad)

Si está activado, se crearán copias de los códigos fuente que guardes con Dev-C++. Este archivo de respaldo se sobrescribirá cada vez que guardes tu código fuente.

Minimizar en la Ejecución del Programa

Si está activado, Dev-C++ se minimizará cuando ejecutes el ejecutable compilado con tus códigos fuente. La ventana de Dev-C++ se restaurará a su posición original cuando termine la ejecución.

Mostrar barras de Herramientas en Pantalla Completa

Por defecto, las barras de herramientas se ocultan cuando Dev-C++ se ejecuta en modo Pantalla Completa. Si está activada, las barras de herramientas se mostrarán siempre.

Hacer doble clic para abrir los archivos del Administrador (Explorador) del proyecto

Si esta opción está activada, necesitarás hacer doble clic en cada uno de los archivos o carpetas del explorador de proyectos para abrirlos en el editor. Si está desactivada bastará con hacer un clic simple.

Auto-Abrir...

Aquí puedes elegir que tus archivos se abran automáticamente cuando el editor abra un proyecto.

Interfaz

Archivos recientes

Aquí puedes especificar el número máximo de archivos que Dev-C++ te mostrará en su menú 'Abrir Archivo Reciente' dentro del menú 'Archivo'.

Idioma

Selecciona el lenguaje de Dev-C++ aquí.

Mostrar Resultados del Compilador

Aquí puedes configurar Dev-C++ para que te muestre los Resultados de Compilación, Depurar,... en una localización (arriba o abajo) determinada en el editor.

Tema, Estilo de los iconos

Selecciona el tema visual de los iconos que quieres que Dev-C++ use.

No mostrar la pantalla de presentación en el inicio

Si está activada, hace que Dev-C++ no muestre su pantalla de presentación con el logo cuando se está cargando.

Usar interfaz con estilo Windows XP

Si está activado, hace que Dev-C++ se muestre con un tema, estilo, parecido al de Windows XP (imágenes, botones, etc.).

Estilo de las ventanas Abir y Guardar

Aquí puedes seleccionar el tipo de diálogo de abrir y guardar archivo que quieres ver.

Directorios

Directorio del programa

Especifica la dirección del programa Dev-C++. También es la localización por defecto de los nuevos proyectos, archivos para abrir, guardar, etc.

Plantillas

Especifica la dirección de las plantillas de proyectos para Dev-C++ (las que se muestran cuando vas a 'Nuevo Proyecto').

Directorio de la biblioteca de iconos

Especifica la dirección de los iconos que vienen con Dev-C++ utilizables para tus proyectos.

Directorio de los archivos de idiomas (traducciones)

Especifica la dirección de los archivos de traducciones de Dev-C++.

Imagen de Presentación

Puedes cambiar aquí la imagen del inicio de Dev-C++ por una alternativa.

Asociación de archivos

Aquí puedes seleccionar uno o más tipos de archivos que Dev-C++ automáticamente asociará con el mismo, y así la próxima vez que hagas clic en un archivo en una ventana de Windows, Dev-C++ lo abrirá automáticamente.

Soporte CVS

Archivo de programa CVS

Introduce el nombre del programa ejecutable para acceder al CVS.

Nivel de compresión

Especifica el nivel de compresión a usar.

Usar SSH en vez de RSH

Usa el programa SSH para conectar.

10.3. Opciones del Editor

10.3.3. Accesos rápidos del teclado en el Editor

Cursor: Estos comandos controlan el comportamiento del cursor.

Comando	Descripción
Fecha izquierda	Un caracter a la izquierda
Fecha derecha	Un caracter a la derecha
Fecha arriba	Una línea arriba
Fecha abajo	Una línea abajo
Control + Izquierda	Una palabra a la izquierda
Control + Derecha	Una palabra a la derecha
Inicio	Inicio de la línea
Fin	Final de la línea
Página Anterior (Re Pág)	Una página arriba (página anterior)
Avanzar Página (Av Pág)	Una página abajo (página siguiente)
Control + Página Anterior (Re pág)	Comienzo de página
Control + Página Siguiente (Av Pág)	Final de página
Control + Inicio	Comienzo del código
Control + Fin	Final del código

Selección: Estos comandos controlan el comportamiento del texto seleccionado.

Comando	Descripción
Shift + Izquierda	Selecciona un caracter a la izquierda
Shift + Derecha	Selecciona un caracter a la derecha
Shift + Arriba	Selecciona una línea hacia arriba
Shift + Abajo	Selecciona una línea hacia abajo
Control + Shift + Izquierda	Selecciona una palabra a la izquierda
Control + Shift + Derecha	Selecciona una palabra a la derecha
Shift + Inicio	Selecciona hasta el inicio de la línea
Shift + Fin	Selecciona hasta el fin de la línea
Shift + Página Arriba	Selecciona una página arriba
Shift + Página Abajo	Selecciona una página abajo
Control + Shift + Página Arriba	Selecciona hasta el comienzo de página
Control + Shift + Página Abajo	Selecciona hasta el fin de la página
Control + Shift + Inicio	Selecciona hasta el inicio del código
Control + Shift + Fin	Selecciona hasta el final del código
Control + A	Selecciona todo
Control + Insert	Copia lo seleccionado al portapapeles
Control + C	

Desplazamiento (Scroll): Estos comandos controlan el comportamiento de todo lo relacionado con el desplazamiento.

Comando	Descripción
Desplazar con Teclas Arriba + Control	Ir una línea arriba
Desplazar con Teclas Abajo + Control	Ir una línea abajo
Desplazar con Tecla Izquierda	Ir a la izquierda un caracter
Desplazar con Tecla Derecha	Ir a la derecha un caracter

Modos: Estos comandos organizan modos.

Comando	Descripción
Insert	Establece modo Sobreescibir / Insertar
Control + Shift + N	Establece el tipo de selección como normal
Control + Shift + C	Establece el tipo de selección de columna
Control + Shift + L	Establece el tipo de selección de línea

Acciones: Estos comandos realizan diversas acciones.

Comando	Descripción
Control + Shift + B	Va al paréntesis o llave concordante
Control + (number)	Se desplaza al marcador (número)
Control + Shift + (number)	Establece el marcador (número)
F1	Ayuda sensitiva según la palabra seleccionada

Suprimir: Todos los comandos que tienen que ver con eliminación y borrado.

Comando	Descripción
Retroceso	Borrar el caracter de la izquierda
Suprimir	Borrar el caracter de la derecha
Control + T	Borrar la palabra de la derecha
Control + Retroceso	Borrar la palabra de la izquierda
Control + Shift + Y	Borra desde el curso hasta final de línea
Control + Y	Borra la línea entera
Enter	Nueva línea
Alt + Retroceso	Deshace si es posible
Control + Z	Deshace si es posible
Alt + Shift + Back	Rehace si es posible
Control + Shift + Z	Rehace si es posible
Shift + Suprimir	Quita el texto seleccionado y lo mueve al portapapeles
Control + X	Quita el texto seleccionado y lo mueve al portapapeles
Shift + Insert	Mueve los contenidos del portapapeles a la posición actual
Control + V	Mueve los contenidos del portapapeles a la posición actual
Control + Shift + I	Mueve el texto seleccionado a la derecha (una tabulación)
Control + Shift + U	Mueve el texto seleccionado a la izquierda (una tabulación)
Tab	Tabulador
Shift + Tab	Tabulador hacia la izquierda

11. Obteniendo ayuda

11.1. Preguntas más frecuentes (FAQ)

¿Por qué no puedo usar las funciones de *conio.h* como por ejemplo `clrsrc()`?

Porque *conio.h* no está incluida en el C estándar. Es una extensión de Borland, y **sólo** funciona con compiladores Borland (y quizás con otros compiladores del mercado). Dev-C++ utiliza GCC, el compilador GNU, como su compilador. GCC es un compilador originario de UNIX, y procura ser portable y estándar.

Si no puedes vivir sin esas funciones, puedes usarlas de esta forma:

Incluye *conio.h* en tu código, y añade el siguiente archivo a tu proyecto: `C:\Dev-Cpp\include\conio.c` (donde `C:\Dev-Cpp` es el nombre del directorio donde tienes Dev-C++ instalado).

Nota también que el soporte de *conio* no es completo.

Mi ventana se cierra al finalizar la ejecución del programa, ¿cómo puedo hacer para que no se cierre y así poder ver los resultados?

Puedes hacerlo de esta forma:

```
#include <stdio.h>
int main(int argc, char **argv)
{
    printf ("Presiona ENTER para continuar.\n");
    getchar (); /* espera la entrada de datos */
    return 0;
}
```

O de esta forma:

```
#include <stdlib.h>
int main(int argc, char **argv)
{
    system ("pause"); /* ejecuta el comando de pausa de MS-DOS */
    return 0;
}
```

Después de construir el proyecto, obtengo un error que dice algo como esto:
`C:\DEV-CPP\LIB\libmingw32.a(main.o)(.text+0x8e): undefined reference to `WinMain@16'`

Probablemente no tengas función `main()` o `WinMain()` declarada en tu programa.

¿Cómo puedo incluir un archivo *.def* para mi DLL?

Ve a Opciones del Proyecto, y en la pestaña de parámetros verás tres cajas. La tercera es la caja del linker. Pon en esa caja: `--def tuarchivo.def`

Tengo problemas extraños en Windows XP

Intenta ejecutar Windows Update y asegúrate de que tienes las actualizaciones de compatibilidad de programas.

¿Cómo puedo activar el modo de Depuración?

Ve a las Opciones del Compilador y haz clic en la pestaña de Configuración. En la sección del Linker, pon "Sí" en "Generar Información para Depurar". Reconstruye todo tu proyecto y ya deberías de poder depurar.

Cuando ejecuto Dev-C++ obtengo un mensaje de error que dice que falta un DLL como 'WININET.DLL', 'MSCVRT.DLL' o 'SHFOLDER.DLL'

Te falta dicho DLL, que debería incluirlo el Windows. Lo puedes descargar del sitio web de Microsoft o desde aquí.

El tamaño del ejecutable es muy elevado, ¿por qué?

Seguramente tengas la opción de "Generar información para depurar" activada. Esta opción incrementa el tamaño del ejecutable para poder depurar el programa. Si quieres que el tamaño sea menor o quieres hacer una versión final de tu programa, ve a "Opciones del Compilador", después haz clic en la pestaña de "Configuración" y en la sección del "Linker", desactiva la opción de "Generar información para depurar".

¿Cómo puedo usar código ensamblador en Dev-C++?

El ensamblador "GNU as" utilice sintaxis AT&T (distinta de la Intel). Aquí se muestra un ejemplo:

```
// 2 variables globales
int AdrIO;
static char ValIO;

void MyFunction(.....)
{
    __asm("mov _AdrIO, %dx") ; // cargando registro 16 bits
    __asm("mov _ValIO, %al") ; // cargando registro 8 bits
    /* No olvidar el guión bajo o underscore _ antes de cada nombre de variable global! */
    __asm("mov %ax,%dx") ; // AX --> DX
}
```

Estoy usando Windows 98 y no puedo compilar

Algunos usuarios han reportado que es necesario aplicar varios parches (actualizaciones) al sistema. Aquí está una lista de ellos, que pueden encontrarse en el sitio de descargas de Microsoft Windows 98 (<http://www.microsoft.com/windows98/downloads/corporate.asp>).

47569us.exe - etiquetado como "Apagado de Windows98SE"

dcom98.exe - mira también esta página:

http://www.microsoft.com/com/dcom/dcom98/dcom1_3.asp

DirectX - este archivo actualizará muchas partes de tu sistema.

(Nota: los parches aquí mencionados son de idiomas inglés británico y estadounidense, intenta bajarte los correspondientes a tu sistema operativo, que seguramente sea Español).

11.2. Listas de correo / Foro

Si tienes dudas cuando estás programando o tienes dudas específicas de Dev-C++, puedes exponer preguntas en el foro o en la lista de correo.

Bloodshed Software / Foro Dev-C++: <http://bloodshed.net/forum>

Dev-C++ Lista de Correo: <http://bloodshed.net/devcpp-ml.html>

11.3. Contactando con los Autores

Sugerencias, correcciones son siempre bienvenidas, pero intenta antes mirar las Preguntas Más Frecuentes FAQ o el Foro de Bloodshed(<http://www.bloodshed.net/forum>) antes de preguntar sobre un problema (el 90 % de los problemas están descritos ahí, y también puedes encontrar la última actualización del FAQ en (<http://www.bloodshed.net/dev/faq.html>)).

Colin Laplace	haiku@bloodshed.net
Hongli Lai	h.lai@chello.nl
Yiannis Mandravellos	mandrav@supergoal.gr
Fernando Muñoz Ledesma (Traductor a español)	ledesmafernando@msn.com http://membres.lycos.fr/ledesma
Sitio Web de Bloodshed Software:	http://www.bloodshed.net
Sitio Web de Mingw:	http://www.mingw.org/