

Comandos GCC y GDB

PROGRAMACIÓN DE SISTEMAS EMBEBIDOS



24 de marzo de 2020

nadia sarahi murguia chavez

ING. MECATRONICA 8vo A

**Comandos GCC.**

GCC. Es un compilador para GNU considerado estándar para los sistemas operativos derivados de UNIX, de código abierto o también de sistemas operativos propietarios, como Mac OS X. Las siglas GCC significa GNU Compiler Colletion (colección de compiladores GNU). Antes estas siglas de GNU C Compiler (Compilador C GNU). Como su nombre indica es una colección de compiladores y admite diversos lenguajes C, C++, Objetive C, Chill, Fortran, Java y Ada.

La primera versión de GCC se hizo en 1987. Esto fue un avance significativo, siendo el primer portátil de la optimización de compilador de C ANSI publicado como software libre. Desde entonces CCG se ha convertido en una de las herramientas más importantes en el desarrollo de software libre. Una revisión importante del compilador llegó con la serie 2.0 en 1992, que añade la capacidad para compilar C + +. En 1997 una rama experimental del compilador (SLGE) se creó para mejorar la optimización y soporte para C + +. A raíz de este trabajo, egcs fue adoptado como la nueva línea principal de desarrollo del CCG, y estas características se hizo ampliamente disponible en la versión 3.0 de GCC en el 2001. Con el tiempo del CCG se ha ampliado para soportar muchos idiomas, incluyendo Fortran, Ada, Java y Objective-C. Su desarrollo se guía por el comité director del CCG, un grupo compuesto por representantes de las comunidades de usuarios del CCG en la industria, la investigación y la academia.

GCC es parte del proyecto GNU, el cual tiene como objetivo mejorar el compilador usado en los sistemas GNU incluyendo la variante GNU/Linux. El desarrollo de GCC usa un entorno de desarrollo abierto y soporta muchas otras plataformas con el fin de fomentar el uso de un compilador-optimizador de clase global, para atraer muchos equipos de desarrollo, para asegurar que GCC y los sistemas GNU funcionen en diferentes arquitecturas y diferentes entornos, y más aún, para extender y mejorar las características de GCC. GCC requiere el conjunto de aplicaciones conocido como binutils para realizar tareas como identificar archivos objeto u obtener su tamaño para copiarlos, traducirlos o crear listas, enlazarlos, o quitarles símbolos innecesarios. El compilador se distribuye bajo la licencia GPL (General Public License) lo que lo hace de libre distribución: se pueden hacer copias de él y regalarlas o venderlas siempre que se incluya el Código fuente (o se indique cómo conseguirlo) y se mantenga la licencia. Existen versiones para prácticamente todos los sistemas operativos. Viene incluido en la mayoría (si no en todas) las distribuciones de GNU/Linux. La versión DOS de este compilador es el DJGPP. En el desarrollo de este compilador participan cientos de voluntarios de todo el mundo.

Estructura.

El proceso de Compilación involucra cuatro etapas sucesivas. Para pasar de un Programa fuente escrito por un humano a un Archivo ejecutable es necesario realizar estas cuatro etapas en forma sucesiva. Los comandos GCC y G++ son capaces de realizar todo el proceso de una sola vez.

* Pre procesado

En esta etapa se interpretan las directivas al Preprocesador. Entre otras cosas, las Variables inicializadas con #define son sustituidas en el código por su valor en todos los lugares donde aparece su nombre.

* Compilación

La Compilación transforma el Código C en el Lenguaje ensamblador propio del Procesador de nuestra máquina.

* Ensamblado

El ensamblado transforma el programa escrito en Lenguaje ensamblador a Código objeto, un Archivo binario en Lenguaje de máquina ejecutable por el Procesador.

* Enlazado

Las Funciones de C/C++ incluidas en nuestro Código, tal como printf () en el ejemplo, se encuentran ya compiladas y ensambladas en Bibliotecas existentes en el sistema. Es preciso incorporar de algún modo el Código binario de estas funciones a nuestro ejecutable. En esto consiste la etapa de enlace, donde se reúnen uno o más módulos en código objeto con el código existente en las bibliotecas.

**Comandos GDB.**

GDB, abreviatura de GNU Debugger, es el depurador más popular para los sistemas UNIX para depurar programas C y C ++. El depurador de GNU, que también se llama GDB, es el depurador más popular para los sistemas UNIX para depurar los programas C y C ++.

GNU Debugger le ayuda a obtener información sobre lo siguiente:

* Si ocurrió un volcado de memoria, ¿en qué declaración o expresión se bloqueó el programa?
* Si se produce un error al ejecutar una función, ¿qué línea del programa contiene la llamada a esa función, y cuáles son los parámetros?
* ¿Cuáles son los valores de las variables del programa en un punto particular durante la ejecución del programa?
* ¿Cuál es el resultado de una expresión particular en un programa?

GDB (depurador de proyectos GNU) es un depurador de base de línea de comandos que es bueno para analizar programas en ejecución y de núcleo. De acuerdo con el manual del usuario, GDB admite C, C ++, D, Go, Objective-C, Fortran, Java, OpenCL C, Pascal, Rust, assembly, Modula-2 y Ada.

GDB tiene el mismo conjunto de funciones que la mayoría de los depuradores, pero es diferente de la mayoría de los que he usado, ya que todo se basa en los comandos de escritura en lugar de hacer clic en los elementos de la GUI. Algunas de estas características incluyen:

* Establecer puntos de ruptura
* Imprimiendo el valor de las variables.
* Establecer el valor de las variables para probar cosas.
* Viendo la pila