



NOMBRE DEL ALUMNO:

Everardo Estrella Rojo

CARRERA:

Ing. Mecatrónica

MATERIA:

Programación de Sistemas Embebidos

GRADO Y GRUPO:

8°-B

CUATRIMESTRE:

Septiembre - Diciembre

NOMBRE DEL DOCENTE:

Carlos Enrique Morán Garabito

¿Qué es un gcc?

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.



Estructura_La interfaz exterior de GCC es generalmente estándar para un sistema UNIX. Los usuarios llaman un programa controlador llamado gcc, que interpreta los argumentos dados, decide que compilador usar para cada archivo y ejecuta el ensamblador con el código resultante, después posiblemente ejecuta el enlazador para producir un programa completo. Cada uno de los compiladores es un programa independiente que toma como entrada Código fuente y produce código en Ensamblador. Todos ellos tienen una



estructura interna común: un Front end por lenguaje que procesa el lenguaje y produce un árbol de sintaxis y un Back end, que convierte esos árboles al Lenguaje RTL (lenguaje de transferencia de registros) de GCC, luego realiza varias optimizaciones y produce el ensamblador utilizando un reconocimiento de patrones específico para la arquitectura, originalmente basado en un Algoritmo de Jack Davidson y Chris Fraser.

Etapas de compilación

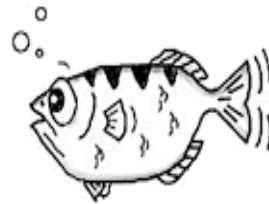
El proceso de Compilación involucra cuatro etapas sucesivas. Para pasar de un Programa fuente escrito por un humano a un Archivo ejecutable es necesario realizar estas cuatro etapas en forma sucesiva. Los comandos Gcc y G++ son capaces de realizar todo el proceso de una sola vez.

1. **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.
2. **Compilación** La Compilación transforma el Código C en el Lenguaje ensamblador propio del Procesador de nuestra máquina.
3. **Ensamblado** El ensamblado transforma el programa escrito en Lenguaje ensamblador a Código objeto, un Archivo binario en Lenguaje de máquina ejecutable por el Procesador.
4. **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.

¿Qué es un comando GDB?

Aquellos que desarrollan en C, conocen de las dificultades a las que se enfrenta cuando trata de depurar un programa, que, por ejemplo, por qué no se agrega un nodo a una lista o por qué no se copia determinado string. GDB (Gnu Project Debugger) es una herramienta que permite entre otras cosas, correr el programa con la posibilidad de detenerlo cuando se cumple cierta condición, avanzar paso a paso, analizar qué ha pasado cuando un programa se detiene o cambiar algunas cosas del programa como el valor de las variables.



GDB
The GNU Project
Debugger

GDB es una herramienta muy poderosa que nos ayudará a encontrar esos errores difíciles, por ejemplo, cuando los punteros no apuntan a donde estamos pensando. Si bien este tutorial está pensado para el lenguaje C, probablemente también sirva para depurar programas en Fortran o C++ con los mismos comandos o similares.

En propias palabras se puede decir que la herramienta de programación gcc es un lenguaje de programación, a continuación, se analizará la estructura principal para hacer un programa ejecutable.

```
(gdb) l 41,51
41     #ifdef TEST
42     int main(int argc, char ** argv){
43         Lista l;
44         puts("Pregunto si la lista sin inicializar es vacía.");
45         printf("%s\n", (l_EsVacía(l)?"Es vacía":"No es vacía"));
46         puts("No es vacía, así que trato de imprimir el contenido del nodo...");
47         printf("%s\n", (char*)l->dato);
48         return 0;
49     }
50     #endif
```

Una prueba de programación utilizando main para poder hacer ejecutable el programa, dicha programación es para realizar una lista con caracteres de "char" comando utilizando para recibir una respuesta tipo letra.

Referencias bibliográficas

- Richard M. Stallman: Using and Porting the GNU Compiler Collection, Free Software Foundation, ISBN 0-595-10035-X Richard
- M. Stallman: Using Gcc: The Gnu Compiler Collection Reference, Free Software Foundation, ISBN 1-882114-39-6 Brian J. Gough: An Introduction to GCC, Network Theory Ltd., ISBN 0-9541617-9-3