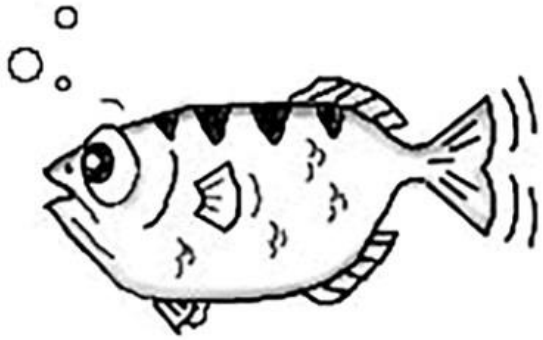
Comando gcc y gbd

ARREOLA VASQUEZ JESUS ALBERTO
T/M
8.-B
ING. MECATRONICA
UPZMG

Comando gcc y gbd

El complemento perfecto de GNU gcc es GNU gdb, para el proceso de debugging. La depuración de software es otro de las etapas esenciales a la hora de crear un programa. Si lo recuerdas, era el proceso para poder identificar y corregir bugs presentes en tu código fuente. No se trata de solventar problemas de sintaxis u otro tipo de errores al tipear el código, sino de errores lógicos.



GDB
The GNU Project
Debugger

GCC

Es un compilador integrado del proyecto GNU para C, C++, Objective C y Fortran; es capaz de recibir un programa fuente en cualquiera de estos lenguajes y generar un programa ejecutable binario en el lenguaje de la máquina donde ha de correr.
La sigla GCC significa "GNU Compiler Collection". Originalmente significaba "GNU C Compiler"; todavía se usa GCC para designar una compilación en C. G++ refiere a una compilación en C++.

Sintaxis.

```
gcc [ opción | archivo ] ...
g++ [ opción | archivo ] ...
```

Las opciones van precedidas de un guion, como es habitual en UNIX, pero las opciones en sí pueden tener varias letras; no pueden agruparse varias opciones tras un mismo guion. Algunas opciones requieren después un nombre de archivo o directorio, otras no. Finalmente, pued

en darse varios nombres de archivo a incluir en el proceso de compilación.

Es decir, cuando creas un código fuente, los errores de sintaxis, cuando olvidas declarar alguna variable, etc., de eso te avisa el propio compilador, que lanzará mensajes de error o advertencias y la compilación fallará. Una vez consigues pasar esa etapa, la siguiente sería la depuración. A pesar de que el compilador haya creado el binario, eso no quiere decir que tu software esté libre de errores. Es posible que se produzcan problemas que solo se den con ciertos eventos puntuales, desbordamientos, etc.

Sufijos en nombres de archivo:

`.c`	fuente en C
`.C .cc .cpp .c++ .cp .cxx`	fuente en C++; se recomienda `.cpp`
`.m`	fuente en Objective-C
`.i`	C preprocesado
`.ii`	C++ preprocesdo
`.s`	fuente en lenguaje ensamblador
`.o`	código objeto
`.h`	archivo para preprocesador (encabezados), no suele figurar en la linea de comando de gcc

Opciones.

- c
Realiza preprocesamiento y compilación, obteniento el archivo en código objeto; no realiza el enlazado.

- E
Realiza solamente el preprocesamiento, enviando el resultado a la salida estándar.

-o archivo
Indica el nombre del archivo de salida, cualesquiera sean las etapas cumplidas.

-Iruta
Especifica la ruta hacia el directorio donde se encuentran los archivos marcados para incluir en el programa fuente. No lleva espacio entre la I y la ruta, así: -I/usr/include

-L
Especifica la ruta hacia el directorio donde se encuentran los archivos de biblioteca con el código objeto de las funciones referenciadas en el programa fuente. No lleva espacio entre la L y la ruta, así: -L/usr/lib

-Wall
Muestra todos los mensajes de error y advertencia del compilador, incluso algunos cuestionables pero en definitiva fáciles de evitar escribiendo el código con cuidado.

-g
Incluye en el ejecutable generado la información necesaria para poder rastrear los errores usando un depurador, tal como GDB (GNU Debugger).

-v
Muestra los comandos ejecutados en cada etapa de compilación y la versión del compilador. Es un informe muy detallado.

Etapas de compilación.

El proceso de compilación involucra cuatro etapas sucesivas: preprocesamiento, compilación, ensamblado y enlazado. 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.

Preprocesado.

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.

GBD

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 que ha pasado cuando un programa se detiene o cambiar algunas cosas del programa como el valor de las variables.

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.

REFERENCIAS:

http://architecnologia.es/programacion-el-depurador-gdb

https://es.scribd.com/document/203199198/Herramientas-GNU