

Manuel Arturo Chinchilla Sánchez - 2013009344

### Resumen #3

Compiling a C++ Program / How the compiler Works / Examining compiled files - An\_Introduction\_to\_GCC-Brian\_Gough

Páginas: 55-61 / 81-86

Para compilar un programa desde la consola utilizamos el comando `$ g++ -Nombre.cc -o nombre`. Con G++ podemos compilar extensiones válidas de C++, como ".cc", ".cpp", ".cxx" o ".C" y el ejecutable resultante de compilado puede ser ejecutado escribiendo simplemente `$ ./Nombre`.

Plantillas: Proveen la posibilidad de definir clases que soportan técnicas genéricas de programación. Cuando una plantilla ya sea de clase o de función es usada con una clase específica o tipo de dato, como float o int, el código de la plantilla en cuestión es compilado con el tipo sustituto. La biblioteca estándar "libstdc++" ofrece un gran rango de plantillas de clases, como listas y colas. Cuando utilizamos plantillas de la biblioteca estándar de C++ no es necesario agregar ninguna opción especial al compilar o ejecutar nuestro programa.

Además de las plantillas proporcionadas por la librería estándar de C++, podemos definir nuestras propias plantillas. La manera recomendada de usar plantillas con g++ es seguir el modelo de inclusión de compilación, donde las definiciones de la plantilla se colocan en la cabeza de los archivos.

Como trabaja el compilador: La secuencia de comandos ejecutados por una sola invocación de GCC consiste en seguir los pasos:

- Pre-procesamiento (para expandir los macros).
- Compilación (a partir del código fuente hasta ensamblador).
- Ensamblador (a partir de lenguaje ensamblador hasta código máquina).
- Enlace (para crear el ejecutable final).

El procesador: Es la primera etapa durante el proceso de compilación, es el encargado de expandir los macros e incluir las cabeceras de los archivos. Para realizar esta etapa el compilador GCC ejecuta el comando:

`$ g++ hello.c > hello.i` (donde hello es el nombre del archivo)

El resultado es un archivo "hello.i" que contiene el código fuente con los macros expandidos. Por convención los archivos procesados tienen la extensión ".i" para los programas en C y ".ii" para los programas en C++

El compilador: La siguiente etapa es la compilación de código fuente a lenguaje de ensamblador. La opción de línea de comando “-S” le dice al gcc que convierta el código fuente procesado a lenguaje de ensamblador:

```
$ gcc -Wall -S hello.i
```

El resultante lenguaje de ensamblador es guardado in el archivo “hello.s”.

El ensamblador: Es el encargado de convertir el lenguaje ensamblado en código máquina. El ensamblador puede ser llamado con el siguiente comando:

```
$ as hello.s -o hello.o
```

El enlazador: Es la etapa final del proceso de compilación, es el enlace de los objetos de archivos para crear el ejecutable. Un ejecutable requiere de varias funciones externas del sistema y del “C run time (crt) libraries”. Para enlazar un programa solo debemos de escribir el siguiente código:

```
$ gcc hello.o
```

Esto vincula el objeto de archivo “hello.o” a la librería estándar de C y produce un archivo ejecutable “a.out”:

```
$ ./a.out
```

```
Hello world!!
```

Identificando Archivos: Cuando el código fuente fue compilado a un archivo o a un ejecutable, la opción usada para compilar ya no son obvias. El comando de archivo se ve en el contenido de un objeto de archivo o ejecutable y determina algunas de sus características, como si fue compilado con enlace dinámico o estático.