**COMNDOS GCC Y GDB**

ING MECATRONICA

8-B

PROGRAMACION DE SISTEMAS EMBEBIDOS

BARAJAS MORALES MARTIN

MORAN GARABITO CARLOS ENRIQUE



Gcc

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.

GCC tiene múltiples frontends, para parsear diferentes lenguajes. Los programas en cada lenguaje pueden ser compilados, o compilados de manera cruzada, para cualquier arquitectura. Por ejemplo, un programa en ADA puede ser compilado para un microcontrolador, o un programa en C para un supercomputador.

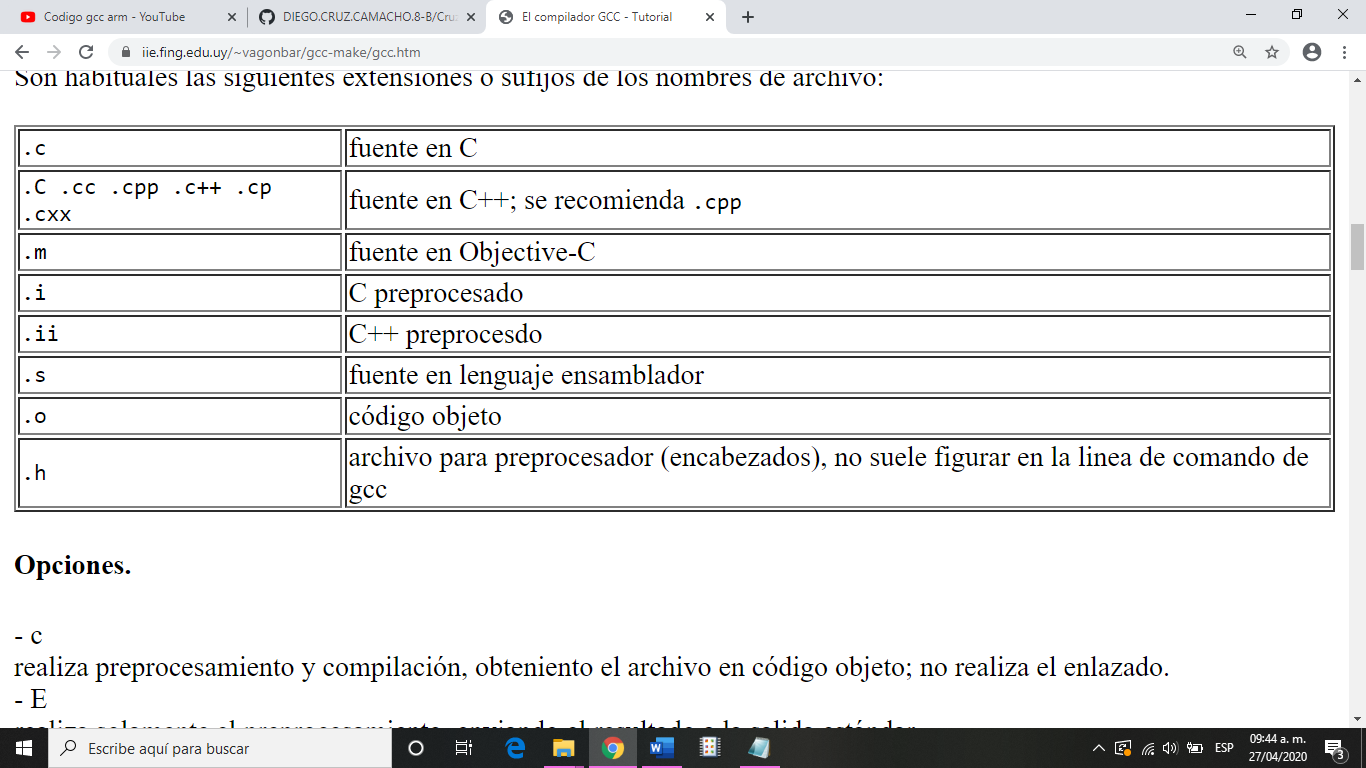
La sigla GCC significa "GNU Compiler Collection". Originalmente significaba "GNU C Compiler"; todavía se usa GCC para designar una compilación en C. G++ refiere a una compilación en C++.

El compilador de C de GNU **gcc** es la aplicación que, dado un conjunto de ficheros de código C, genera un programa ejecutable. Como el resto de aplicaciones, se puede invocar desde el intérprete de comandos.

Como todo comando, **gcc**también tiene su página de manual en donde se explica su funcionamiento, así como las opciones con las que se puede modificar su comportamiento. Con el comando siguiente se podrá observar:



Extensiones o sufijos de los nombres de archivo:



La opción ‘-c’ es usada para compilar un fichero fuente para crear un fichero objeto. Por ejemplo, el siguiente comando compilara el fichero fuente ‘main.es.c’ para generar un fichero objeto: $ gcc -Wall -c main.es.c Esto produce un fichero objeto ‘main.o’ que contiene el código máquina para la función main. Este contiene una referencia la función externa hola, pero la correspondiente dirección de memoria se deja indefinida en el fichero objeto en esta fase (se introducirá después al enlazarse).

El correspondiente comando para compilar la función hola en el código fuente ‘hola\_fn.es.c’ es: $ gcc -Wall -c hola\_fn.es.c Esto produce el fichero objeto ‘hola\_fn.o’

Para usar el makefile, se escribe make. Al llamarse sin argumentos, se construye el primer objetivo en el makefile, produciendo el ejecutable ‘main’.

$ make

gcc -Wall -c -o main.o main.es.c

gcc -Wall -c -o hola\_fn.o hola\_fn.es.c

gcc main.o hola\_fn.o -o main

$ ./main

¡Hola, mundo!

Para reconstruir el ejecutable después de modificar un fichero fuente, simplemente se escribe make de nuevo. Al comprobar las fechas de modificación de los objetivos y de los ficheros dependientes, make identifica los ficheros que han cambiado y regenera los ficheros intermedios correspondientes y necesarios para actualizar los objetivos:

$ emacs main.es.c (editar el fichero) $

make gcc -Wall -c -o main.o main.es.c

gcc main.o hola\_fn.o -o main

$ ./main

¡Hola, cualquiera!

Finalmente, para eliminar los ficheros generados, se escribe make clean:

$ make clean

rm -f main main.o hola\_fn.o

Gdb

Un depurador es una herramienta de programación que nos permite la detección y corrección de errores de ejecución y errores lógicos. Este tipo de errores aparecen a partir del momento en el que tenemos un código sintácticamente correcto del que obtenemos un programa ejecutable. Es al probar este programa ejecutable cuando se puede detectar la aparición de errores de ejecución y lógicos.

La depuración de programas es una tarea bastante complicada, sobre todo con depuradores en línea como gdb, por lo que para facilitar esta tarea los programas deben estar bien especificados y diseñados y estar escritos con la máxima claridad (legibilidad de los programas). La depuración de un programa se facilita si es correcta su especificación, diseño y legibilidad y nunca a la inversa.

Un depurador permite la ejecución controlada del programa de forma que se puede establecer una relación entre el código fuente y el código que se está ejecutando. De esta forma el depurador puede indicar a que línea del código fuente corresponde la instrucción (o instrucciones) de código ejecutable que en ese momento se está ejecutando.

Para depurar un programa se debe ejecutar el depurador al que se le pasa como parámetro el nombre del ejecutable del programa que queremos depurar