

## Taller #2

Instituto Tecnológico de Costa Rica  
Introducción a los Sistemas Embebidos - CE 5303  
Prof. Luis Alonso Barboza Artavia

**Estudiante:** Esteban Josué Alvarado Vargas

**Carné:** 2018109336

## GCC

### Pregunta 1.1

¿Qué es GCC?

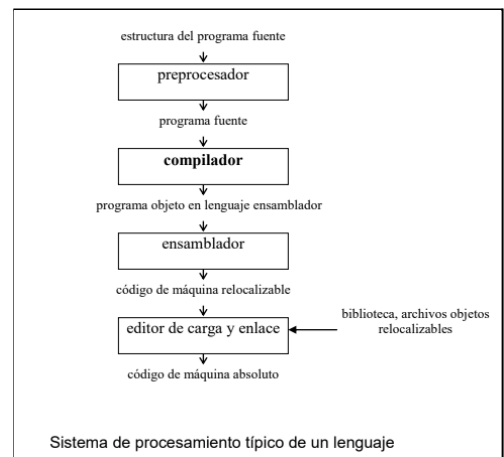
**GCC** (*GNU Compiler Collection*) es un compilador integrado para C, C++, Objective C y Fortran. Es capaz de recibir un código fuente en cualquiera de estos lenguajes y generar su programa ejecutable para la arquitectura de la máquina que lo ejecutará [1].

### Pregunta 1.2

¿Cuáles son las 4 etapas de compilación?

Un **compilador** es un programa que convierte un código fuente escrito en algún lenguaje de alto nivel a lenguaje máquina. El programa fuente se conoce como programa objeto y es guardado en forma de archivo ejecutable. Las etapas del proceso de compilación son [2]:

1. Preprocesador
2. Compilador
3. Ensamblador
4. Editor de carga y enlace



### Pregunta 1.3

¿Qué comando deberá emplear para generar el código en ensamblador de un archivo fuente, por ejemplo, `calculadora.c`?

El comando que debe ejecutarse es: `gcc -S calculadora.c` [3]

### Pregunta 1.4

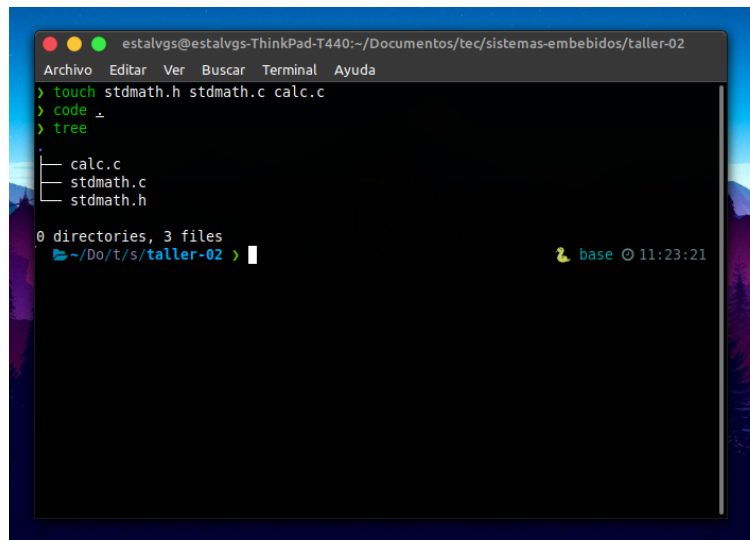
¿Cuál es la diferencia entre biblioteca estática y una dinámica?

Una **biblioteca estática** es aquella que se "copia" en un programa al compilarse, solo se copia aquella parte de la biblioteca que el programa necesita. Por otra parte, una **biblioteca dinámica** es una que ya ha sido compilada y no se copia en un programa al compilarse. Cuando el programa se ejecute, cada vez que el código necesite algo de la biblioteca, irá a buscarlo a esta [4].

## Pregunta 2.1

Debe desarrollar una biblioteca, en lenguaje C<sup>1</sup>, la cual ofrecerá seis funciones matemáticas: suma, resta, multiplicación, división, raíz cuadrada y coseno (por aproximación). Para el desarrollo de la biblioteca, tome en cuenta los siguientes aspectos:

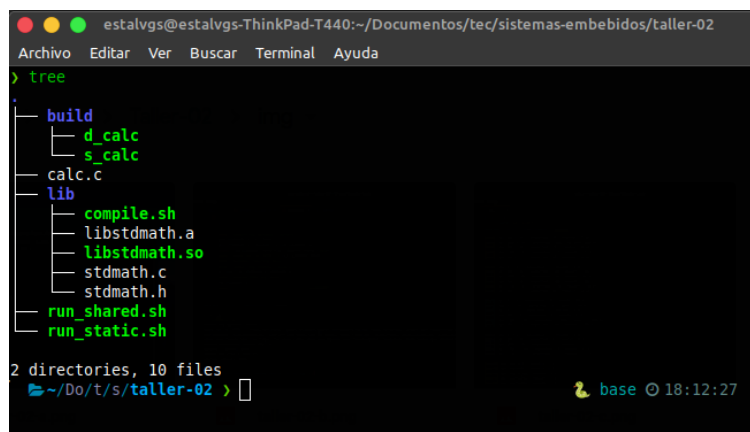
- Debe crear un archivo de cabecera `biblioteca.h`, que contenga únicamente la definición de las funciones a utilizar y las variables globales que requiera. Debe crear además el archivo `biblioteca.c`, donde desarrolle cada una de las funciones.



```
estalvgs@estalvgs-ThinkPad-T440:~/Documentos/tec/sistemas-embebidos/taller-02
Archivo Editar Ver Buscar Terminal Ayuda
> touch stdmath.h stdmath.c calc.c
> code
> tree
.
├── calc.c
├── stdmath.c
└── stdmath.h

0 directories, 3 files
~/Do/t/s/taller-02 >
```

- La biblioteca deberá ser creada tanto estática como dinámicamente, es decir, al finalizar el ejercicio deberá tener dos archivos: `libbiblioteca.a` y `libbiblioteca.so`

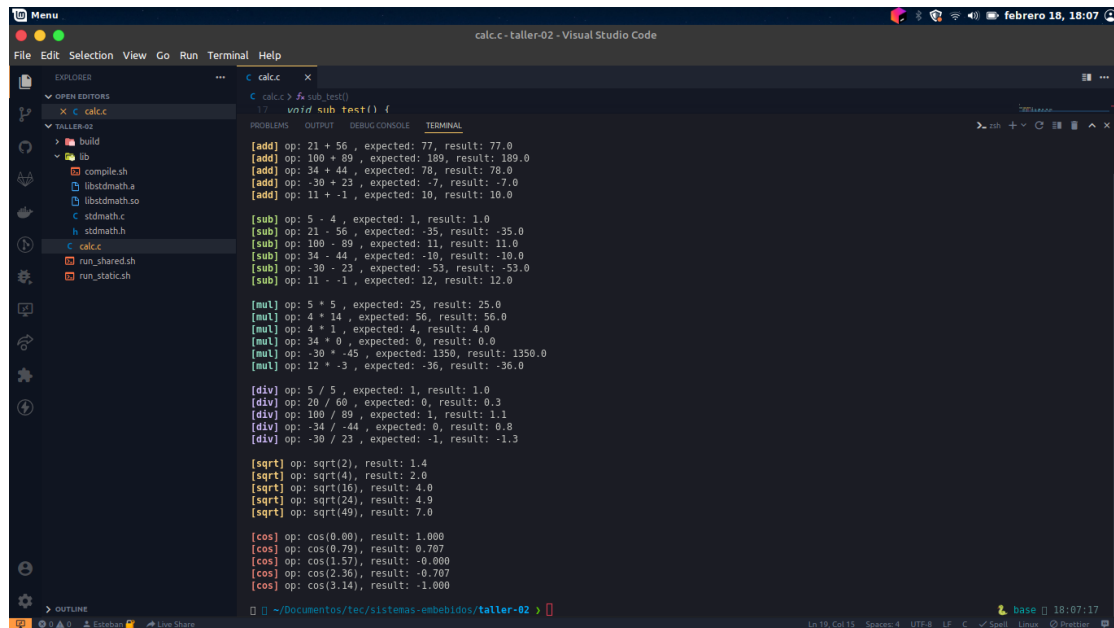


```
estalvgs@estalvgs-ThinkPad-T440:~/Documentos/tec/sistemas-embebidos/taller-02
Archivo Editar Ver Buscar Terminal Ayuda
> tree
.
├── build
│   ├── d_calc
│   └── s_calc
├── calc.c
├── lib
│   ├── compile.sh
│   ├── libstdmath.a
│   ├── libstdmath.so
│   ├── stdmath.c
│   └── stdmath.h
├── run_shared.sh
└── run_static.sh

2 directories, 10 files
~/Do/t/s/taller-02 >
```

<sup>1</sup> Los archivos del código fuente se encuentran en el directorio taller-02

→ Por último, debe crear un archivo `calculadora.c` y su correspondiente ejecutable, para verificar el correcto funcionamiento de ambas bibliotecas.



```
17 void sub_test() {
[add] op: 21 + 56 , expected: 77, result: 77.0
[add] op: 100 + 89 , expected: 189, result: 189.0
[add] op: 34 + 44 , expected: 78, result: 78.0
[add] op: -30 + 23 , expected: -7, result: -7.0
[add] op: 11 + -1 , expected: 10, result: 10.0

[sub] op: 5 - 4 , expected: 1, result: 1.0
[sub] op: 21 - 56 , expected: -35, result: -35.0
[sub] op: 100 - 89 , expected: 11, result: 11.0
[sub] op: 34 - 44 , expected: -10, result: -10.0
[sub] op: -30 - 23 , expected: -53, result: -53.0
[sub] op: 11 - -1 , expected: 12, result: 12.0

[mul] op: 5 * 5 , expected: 25, result: 25.0
[mul] op: 4 * 14 , expected: 56, result: 56.0
[mul] op: 4 * 1 , expected: 4, result: 4.0
[mul] op: 34 * 0 , expected: 0, result: 0.0
[mul] op: -30 * -45 , expected: 1350, result: 1350.0
[mul] op: 12 * -3 , expected: -36, result: -36.0

[div] op: 5 / 5 , expected: 1, result: 1.0
[div] op: 20 / 60 , expected: 0, result: 0.3
[div] op: 100 / 89 , expected: 1, result: 1.1
[div] op: -34 / -44 , expected: 0, result: 0.8
[div] op: -30 / 23 , expected: -1, result: -1.3

[sqrt] op: sqrt(2), result: 1.4
[sqrt] op: sqrt(4), result: 2.0
[sqrt] op: sqrt(16), result: 4.0
[sqrt] op: sqrt(24), result: 4.9
[sqrt] op: sqrt(49), result: 7.0

[cos] op: cos(0.00), result: 1.000
[cos] op: cos(0.79), result: 0.707
[cos] op: cos(1.57), result: -0.000
[cos] op: cos(2.36), result: -0.707
[cos] op: cos(3.14), result: -1.000
```

## Referencias

- [1]H. usuarios and S. instalado, "GCC | Documentación SCAYLE", Scayle.es. [Online]. Available: <https://www.scayle.es/manual/es/hpc/software-instalado/gcc>. [Accessed: 08- Apr- 2022].
- [2]J. Águila, Fases de un compilador. Universidad de Magallanes, 2004.
- [3]"Option Summary (Using the GNU Compiler Collection (GCC))", Gcc.gnu.org. [Online]. Available: <https://gcc.gnu.org/onlinedocs/gcc-11.2.0/gcc/Option-Summary.html#Option-Summary>. [Accessed: 08- Apr- 2022].
- [4]"Librerías estáticas y dinámicas", Chuidiang.org, 2007. [Online]. Available: <http://www.chuidiang.org/clinix/herramientas/librerias.php>. [Accessed: 08- Apr- 2022].