

# Informática II

## El compilador de C del proyecto GNU (gcc, g++) – Bibliotecas

Gonzalo F. Perez Paina



Universidad Tecnológica Nacional  
Facultad Regional Córdoba  
UTN-FRC

# Enlazando con bibliotecas externas

## Biblioteca

Colección de archivos **objetos** precompilados que pueden ser enlazados dentro de un programas.

# Enlazando con bibliotecas externas

## Biblioteca

Colección de archivos **objetos** precompilados que pueden ser enlazados dentro de un programas.

**Estática:** Archivos especiales con la extensión **.a** (ejemplo: **libm.a**)

**Dinámica:** Archivos especiales con la extensión **.so** (ejemplo: **libm.so**)

# Enlazando con bibliotecas externas

## Biblioteca

Colección de archivos **objetos** precompilados que pueden ser enlazados dentro de un programas.

**Estática:** Archivos especiales con la extensión **.a** (ejemplo: **libm.a**)

**Dinámica:** Archivos especiales con la extensión **.so** (ejemplo: **libm.so**)

- ▶ Se encuentran normalmente en directorios
  - ▶ **/usr/lib** o **/lib**
  - ▶ **/usr/lib64** o **/lib64**
- ▶ o en directorios específicos de la arquitectura
  - ▶ **/usr/lib/i386-linux-gnu/**
  - ▶ **/usr/lib/x86\_64-linux-gnu**

# Enlazando con bibliotecas externas

## Biblioteca

Colección de archivos **objetos** precompilados que pueden ser enlazados dentro de un programas.

**Estática:** Archivos especiales con la extensión **.a** (ejemplo: **libm.a**)

**Dinámica:** Archivos especiales con la extensión **.so** (ejemplo: **libm.so**)

- ▶ Se encuentran normalmente en directorios
  - ▶ **/usr/lib** o **/lib**
  - ▶ **/usr/lib64** o **/lib64**
- ▶ o en directorios específicos de la arquitectura
  - ▶ **/usr/lib/i386-linux-gnu/**
  - ▶ **/usr/lib/x86\_64-linux-gnu**

Las declaraciones de los **prototipos de funciones** de las funciones de biblioteca se encuentran en archivos de cabecera (archivo con extensión **.h**)

# Enlazando con bibliotecas externas – Ejemplo

calc.c

---

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main(void)
5 {
6     double x = 2.0;
7     double y = sqrt(x);
8     printf("La raíz cuadrada de %f es %f\n", x, y);
9     return 0;
10 }
```

---

# Enlazando con bibliotecas externas – Ejemplo

calc.c

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main(void)
5 {
6     double x = 2.0;
7     double y = sqrt(x);
8     printf("La raíz cuadrada de %f es %f\n", x, y);
9     return 0;
10 }
```

```
> gcc -Wall -std=c90 calc.c -o calc
/tmp/ccelc3rZ.o: In function `main':
calc.c:(.text+0x23): undefined reference to `sqrt'
collect2: error: ld returned 1 exit status
```

# Enlazando con bibliotecas externas – Ejemplo

calc.c

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main(void)
5 {
6     double x = 2.0;
7     double y = sqrt(x);
8     printf("La raíz cuadrada de %f es %f\n", x, y);
9     return 0;
10 }
```

```
> gcc -Wall -std=c90 calc.c -o calc
/tmp/cceIc3rZ.o: In function `main':
calc.c:(.text+0x23): undefined reference to `sqrt'
collect2: error: ld returned 1 exit status
```

► /tmp/cceIc3rZ.o



# Enlazando con bibliotecas externas – Ejemplo

calc.c

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main(void)
5 {
6     double x = 2.0;
7     double y = sqrt(x);
8     printf("La raíz cuadrada de %f es %f\n", x, y);
9     return 0;
10 }
```

```
> gcc -Wall -std=c90 calc.c -o calc
/tmp/cceIc3rZ.o: In function `main':
calc.c:(.text+0x23): undefined reference to `sqrt'
collect2: error: ld returned 1 exit status
```

- ▶ /tmp/cceIc3rZ.o
- ▶ undefined reference (.text+0x23)

# Enlazando con bibliotecas externas – Ejemplo

calc.c

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main(void)
5 {
6     double x = 2.0;
7     double y = sqrt(x);
8     printf("La raíz cuadrada de %f es %f\n", x, y);
9     return 0;
10 }
```

```
> gcc -Wall -std=c90 calc.c -o calc
/tmp/cceIc3rZ.o: In function `main':
calc.c:(.text+0x23): undefined reference to `sqrt'
collect2: error: ld returned 1 exit status
```

- ▶ /tmp/cceIc3rZ.o
- ▶ undefined reference (.text+0x23)
- ▶ ld returned 1

# Enlazando con bibliotecas externas – Ejemplo

calc.c

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main(void)
5 {
6     double x = 2.0;
7     double y = sqrt(x);
8     printf("La raíz cuadrada de %f es %f\n", x, y);
9     return 0;
10 }
```

```
> gcc -Wall -std=c90 calc.c -o calc
/tmp/cceIc3rZ.o: In function `main':
calc.c:(.text+0x23): undefined reference to `sqrt'
collect2: error: ld returned 1 exit status
```

- ▶ /tmp/cceIc3rZ.o
- ▶ undefined reference (.text+0x23)
- ▶ ld returned 1
- ▶ Borrar el include a math.h y re-compilar (gcc -c)

# Enlazando con bibliotecas externas – Ejemplo

✓ Enlazado con biblioteca estática (buscar `libm.a`)

# Enlazando con bibliotecas externas – Ejemplo

## ✓ Enlazado con biblioteca estática (buscar `libm.a`)

```
> gcc -static -Wall calc.c /usr/lib/x86_64-linux-gnu/libm.a\  
-o calc_static
```

# Enlazando con bibliotecas externas – Ejemplo

## ✓ Enlazado con biblioteca estática (buscar `libm.a`)

```
> gcc -static -Wall calc.c /usr/lib/x86_64-linux-gnu/libm.a\  
-o calc_static
```

(Notar el flag `-static`)

# Enlazando con bibliotecas externas – Ejemplo

## ✓ Enlazado con biblioteca estática (buscar `libm.a`)

```
> gcc -static -Wall calc.c /usr/lib/x86_64-linux-gnu/libm.a\  
-o calc_static
```

(Notar el flag `-static`)

## ✓ Enlazado con biblioteca dinámica

```
> gcc -Wall calc.c -lm -o calc_dynamic
```

# Enlazando con bibliotecas externas – Ejemplo

## ✓ Enlazado con biblioteca estática (buscar `libm.a`)

```
> gcc -static -Wall calc.c /usr/lib/x86_64-linux-gnu/libm.a\  
-o calc_static
```

(Notar el flag `-static`)

## ✓ Enlazado con biblioteca dinámica

```
> gcc -Wall calc.c -lm -o calc_dynamic
```

`-lNAME` enlaza contra la biblioteca `libNAME.so` (p.e. `libm.so`).



# Enlazando con bibliotecas externas – Ejemplo

## ✓ Enlazado con biblioteca estática (buscar `libm.a`)

```
> gcc -static -Wall calc.c /usr/lib/x86_64-linux-gnu/libm.a\  
-o calc_static
```

(Notar el flag `-static`)

## ✓ Enlazado con biblioteca dinámica

```
> gcc -Wall calc.c -lm -o calc_dynamic
```

`-lNAME` enlaza contra la biblioteca `libNAME.so` (p.e. `libm.so`).

- ▶ Probar comando `ldd` y `file` con ambas aplicaciones.
- ▶ Ver tamaños de los binario (`ls -lh`).

# Archivos de cabecera de bibliotecas

badconv.c

---

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     double x = strtod("123", NULL);
6     printf("El valor es %f\n", x);
7     return 0;
8 }
```

---

# Archivos de cabecera de bibliotecas

## badconv.c

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     double x = strtod("123", NULL);
6     printf("El valor es %f\n", x);
7     return 0;
8 }
```

## Compilar

```
gcc -Wall badconv.c -o badconv
badconv.c: In function 'main':
badconv.c:5:14: warning: implicit declaration of function
      'strtod' [-Wimplicit-function-declaration]
      double x = strtod("123", NULL);
                  ^
```

# Archivos de cabecera de bibliotecas

## badconv.c

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     double x = strtod("123", NULL);
6     printf("El valor es %f\n", x);
7     return 0;
8 }
```

## Compilar

```
gcc -Wall badconv.c -o badconv
badconv.c: In function 'main':
badconv.c:5:14: warning: implicit declaration of function
      'strtod' [-Wimplicit-function-declaration]
      double x = strtod("123", NULL);
                  ^
```

## Ejecutar

```
> ./badconv
El valor es 0.000000
```

# Archivos de cabecera de bibliotecas

badconv.c

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     double x = strtod("123", NULL);
6     printf("El valor es %f\n", x);
7     return 0;
8 }
```

## Compilar

```
gcc -Wall badconv.c -o badconv
badconv.c: In function 'main':
badconv.c:5:14: warning: implicit declaration of function
      'strtod' [-Wimplicit-function-declaration]
      double x = strtod("123", NULL);
                  ^
```

## Ejecutar

```
> ./badconv
El valor es 0.000000
```

Agregar archivo de cabecera y probar nuevamente.

# Otras opciones de compilación

`ERROR: FILE.h: No such file or directory`

# Otras opciones de compilación

**ERROR:** FILE.h: No such file or directory

Qué significa?

# Otras opciones de compilación

**ERROR:** FILE.h: No such file or directory

Qué significa? No se encuentra un archivo de cabecera de una biblioteca



# Otras opciones de compilación

**ERROR:** FILE.h: No such file or directory

Qué significa? No se encuentra un archivo de cabecera de una biblioteca

**ERROR:** /usr/bin/ld: cannot find library

# Otras opciones de compilación

**ERROR:** FILE.h: No such file or directory

Qué significa? No se encuentra un archivo de cabecera de una biblioteca

**ERROR:** /usr/bin/ld: cannot find library

Qué significa? No se encuentra la biblioteca compartida al enlazar

# Otras opciones de compilación

**ERROR:** FILE.h: No such file or directory

Qué significa? No se encuentra un archivo de cabecera de una biblioteca

**ERROR:** /usr/bin/ld: cannot find library

Qué significa? No se encuentra la biblioteca compartida al enlazar

Por defecto, gcc busca los archivos en los siguientes directorios:

## Archivos de cabecera:

- ▶ /usr/include
- ▶ /usr/local/include

# Otras opciones de compilación

**ERROR:** FILE.h: No such file or directory

Qué significa? No se encuentra un archivo de cabecera de una biblioteca

**ERROR:** /usr/bin/ld: cannot find library

Qué significa? No se encuentra la biblioteca compartida al enlazar

Por defecto, gcc busca los archivos en los siguientes directorios:

## Archivos de cabecera:

- ▶ /usr/include
- ▶ /usr/local/include

## Bibliotecas:

- ▶ /usr/lib
- ▶ /usr/local/lib

# Otras opciones de compilación

**ERROR:** FILE.h: No such file or directory

Qué significa? No se encuentra un archivo de cabecera de una biblioteca

**ERROR:** /usr/bin/ld: cannot find library

Qué significa? No se encuentra la biblioteca compartida al enlazar

Por defecto, gcc busca los archivos en los siguientes directorios:

## Archivos de cabecera:

- ▶ /usr/include
- ▶ /usr/local/include

## Bibliotecas:

- ▶ /usr/lib
- ▶ /usr/local/lib

Opciones de compilación para agregar rutas: -I y -L. Ejemplos:

```
> gcc -I/usr/include/newlib source.c
```

```
> gcc -L/usr/X11/lib -lX11 source.c
```

# Otras opciones de compilación

**ERROR:** FILE.h: No such file or directory

Qué significa? No se encuentra un archivo de cabecera de una biblioteca

**ERROR:** /usr/bin/ld: cannot find library

Qué significa? No se encuentra la biblioteca compartida al enlazar

Por defecto, gcc busca los archivos en los siguientes directorios:

## Archivos de cabecera:

- ▶ /usr/include
- ▶ /usr/local/include

## Bibliotecas:

- ▶ /usr/lib
- ▶ /usr/local/lib

Opciones de compilación para agregar rutas: -I y -L. Ejemplos:

```
> gcc -I/usr/include/newlib source.c
```

```
> gcc -L/usr/X11/lib -lX11 source.c
```

- ▶ -IPATH: -I/usr/include/libusb, -I\$HOME/milib/include/, etc.
- ▶ -LPATH: -L/usr/lib/graphivz, -L\$HOME/milib/lib, etc.



# Estándares del lenguaje C

`gcc` compila por defecto el dialecto de GNU del lenguaje C, llamado GNU C.



# Estándares del lenguaje C

`gcc` compila por defecto el dialecto de GNU del lenguaje C, llamado GNU C.

Este dialecto incorpora el estándar oficial ANSI/ISO con varias extensiones<sup>1</sup> útiles para sistemas GNU, por ejemplo:

---

<sup>1</sup><https://gcc.gnu.org/onlinedocs/gcc/C-Extensions.html>

# Estándares del lenguaje C

gcc compila por defecto el dialecto de GNU del lenguaje C, llamado GNU C.

Este dialecto incorpora el estándar oficial ANSI/ISO con varias extensiones<sup>1</sup> útiles para sistemas GNU, por ejemplo:

- ▶ funciones definidas dentro de funciones

---

<sup>1</sup><https://gcc.gnu.org/onlinedocs/gcc/C-Extensions.html>

# Estándares del lenguaje C

gcc compila por defecto el dialecto de GNU del lenguaje C, llamado GNU C.

Este dialecto incorpora el estándar oficial ANSI/ISO con varias extensiones<sup>1</sup> útiles para sistemas GNU, por ejemplo:

- ▶ funciones definidas dentro de funciones
- ▶ vectores de tamaño variable

---

<sup>1</sup><https://gcc.gnu.org/onlinedocs/gcc/C-Extensions.html>

# Estándares del lenguaje C

gcc compila por defecto el dialecto de GNU del lenguaje C, llamado GNU C.

Este dialecto incorpora el estándar oficial ANSI/ISO con varias extensiones<sup>1</sup> útiles para sistemas GNU, por ejemplo:

- ▶ funciones definidas dentro de funciones
- ▶ vectores de tamaño variable
- ▶ estructuras vacías, etc.

---

<sup>1</sup><https://gcc.gnu.org/onlinedocs/gcc/C-Extensions.html>

# Estándares del lenguaje C

gcc compila por defecto el dialecto de GNU del lenguaje C, llamado GNU C.

Este dialecto incorpora el estándar oficial ANSI/ISO con varias extensiones<sup>1</sup> útiles para sistemas GNU, por ejemplo:

- ▶ funciones definidas dentro de funciones
- ▶ vectores de tamaño variable
- ▶ estructuras vacías, etc.

Opciones más comunes:

- ▶ **-ansi**: en C es equivalente a **-std=c90**

---

<sup>1</sup><https://gcc.gnu.org/onlinedocs/gcc/C-Extensions.html>

# Estándares del lenguaje C

gcc compila por defecto el dialecto de GNU del lenguaje C, llamado GNU C.

Este dialecto incorpora el estándar oficial ANSI/ISO con varias extensiones<sup>1</sup> útiles para sistemas GNU, por ejemplo:

- ▶ funciones definidas dentro de funciones
- ▶ vectores de tamaño variable
- ▶ estructuras vacías, etc.

Opciones más comunes:

- ▶ `-ansi`: en C es equivalente a `-std=c90`
- ▶ `-Wall`: habilita todas las advertencias (warnings)

---

<sup>1</sup><https://gcc.gnu.org/onlinedocs/gcc/C-Extensions.html>

# Estándares del lenguaje C

gcc compila por defecto el dialecto de GNU del lenguaje C, llamado GNU C.

Este dialecto incorpora el estándar oficial ANSI/ISO con varias extensiones<sup>1</sup> útiles para sistemas GNU, por ejemplo:

- ▶ funciones definidas dentro de funciones
- ▶ vectores de tamaño variable
- ▶ estructuras vacías, etc.

Opciones más comunes:

- ▶ `-ansi`: en C es equivalente a `-std=c90`
- ▶ `-Wall`: habilita todas las advertencias (warnings)
- ▶ `-Werror`: convierte las advertencias en errores

---

<sup>1</sup><https://gcc.gnu.org/onlinedocs/gcc/C-Extensions.html>

# Estándares del lenguaje C

gcc compila por defecto el dialecto de GNU del lenguaje C, llamado GNU C.

Este dialecto incorpora el estándar oficial ANSI/ISO con varias extensiones<sup>1</sup> útiles para sistemas GNU, por ejemplo:

- ▶ funciones definidas dentro de funciones
- ▶ vectores de tamaño variable
- ▶ estructuras vacías, etc.

Opciones más comunes:

- ▶ `-ansi`: en C es equivalente a `-std=c90`
- ▶ `-Wall`: habilita todas las advertencias (warnings)
- ▶ `-Werror`: convierte las advertencias en errores
- ▶ `-pedantic`: genera advertencias si se utiliza alguna extensión de GNU

---

<sup>1</sup><https://gcc.gnu.org/onlinedocs/gcc/C-Extensions.html>



# Estándares del lenguaje C

gcc compila por defecto el dialecto de GNU del lenguaje C, llamado GNU C.

Este dialecto incorpora el estándar oficial ANSI/ISO con varias extensiones<sup>1</sup> útiles para sistemas GNU, por ejemplo:

- ▶ funciones definidas dentro de funciones
- ▶ vectores de tamaño variable
- ▶ estructuras vacías, etc.

Opciones más comunes:

- ▶ `-ansi`: en C es equivalente a `-std=c90`
- ▶ `-Wall`: habilita todas las advertencias (warnings)
- ▶ `-Werror`: convierte las advertencias en errores
- ▶ `-pedantic`: genera advertencias si se utiliza alguna extensión de GNU
- ▶ `-std`: puede ser `-std=c90`, `-std=c99`, etc.

---

<sup>1</sup><https://gcc.gnu.org/onlinedocs/gcc/C-Extensions.html>

# Estándares del lenguaje C – ANSI/ISO estricto

gnucarray.c

---

```
1 #include <stdio.h>
2
3 int main(int argc, char *argv[])
4 {
5     int i, n = argc;
6     double x[n];
7
8     for(i = 0; i < n; i++)
9         x[i] = i;
10
11     return 0;
12 }
```

---

Ejemplo de programa con array de tamaño variable de la extensión de GNU C.

# Estándares del lenguaje C – ANSI/ISO estricto

gnucarray.c

---

```
1 #include <stdio.h>
2
3 int main(int argc, char *argv[])
4 {
5     int i, n = argc;
6     double x[n];
7
8     for(i = 0; i < n; i++)
9         x[i] = i;
10
11     return 0;
12 }
```

---

Ejemplo de programa con array de tamaño variable de la extensión de GNU C.

```
> gcc -Wall -ansi gnuarray.c
```

# Estándares del lenguaje C – ANSI/ISO estricto

gnucarray.c

---

```
1 #include <stdio.h>
2
3 int main(int argc, char *argv[])
4 {
5     int i, n = argc;
6     double x[n];
7
8     for(i = 0; i < n; i++)
9         x[i] = i;
10
11     return 0;
12 }
```

---

Ejemplo de programa con array de tamaño variable de la extensión de GNU C.

```
> gcc -Wall -ansi gnuarray.c
```

```
> gcc -Wall -ansi -pedantic gnuarray.c
gnuarray.c: In function 'main':
gnuarray.c:6:3: warning: ISO C90 forbids variable length
      array 'x' [-Wvla]
```



# Construcción de bibliotecas

Se creará una biblioteca pequeña `libsaludo` que contiene dos funciones `hola` y `adios`.

# Construcción de bibliotecas

Se creará una biblioteca pequeña `libsaludo` que contiene dos funciones `hola` y `adios`.

`saludo.h`

---

```
1 #ifndef SALUDO_H
2 #define SALUDO_H
3
4 void hola(const char * );
5 void adios(void);
6
7 #endif
```

---

# Construcción de bibliotecas

Se creará una biblioteca pequeña `libsaludo` que contiene dos funciones `hola` y `adios`.

## saludo.h

---

```
1 #ifndef SALUDO_H
2 #define SALUDO_H
3
4 void hola(const char * );
5 void adios(void);
6
7 #endif
```

---

## hola.c

---

```
1 #include <stdio.h>
2 #include "saludo.h"
3
4 void hola(const char * nombre)
5 {
6     printf("Hola, %s!\n", nombre);
7 }
```

---



# Construcción de bibliotecas

Se creará una biblioteca pequeña `libsaludo` que contiene dos funciones `hola` y `adios`.

## saludo.h

```
1 #ifndef SALUDO_H
2 #define SALUDO_H
3
4 void hola(const char * );
5 void adios(void);
6
7 #endif
```

## hola.c

```
1 #include <stdio.h>
2 #include "saludo.h"
3
4 void hola(const char * nombre)
5 {
6     printf("Hola, %s!\n", nombre);
7 }
```

## adios.c

```
1 #include <stdio.h>
2 #include "saludo.h"
3
4 void adios(void)
5 {
6     printf("Adios!\n");
7 }
```

# Construcción de bibliotecas – Estática

## Compilar

```
> gcc -Wall -c hola.c  
> gcc -Wall -c adios.c
```

# Construcción de bibliotecas – Estática

## Compilar

```
> gcc -Wall -c hola.c  
> gcc -Wall -c adios.c
```

## Combinar los archivos y generar la biblioteca

```
> ar cr libsaludo.a hola.o adios.o
```

# Construcción de bibliotecas – Estática

## Compilar

```
> gcc -Wall -c hola.c  
> gcc -Wall -c adios.c
```

## Combinar los archivos y generar la biblioteca

```
> ar cr libsaludo.a hola.o adios.o
```

- ▶ **c**: crear el archivo (\*.a)
- ▶ **r**: insertar un miembro (reemplazándolo si existe)

# Construcción de bibliotecas – Estática

## Compilar

```
> gcc -Wall -c hola.c  
> gcc -Wall -c adios.c
```

## Combinar los archivos y generar la biblioteca

```
> ar cr libsaludo.a hola.o adios.o
```

- ▶ **c**: crear el archivo (\*.a)
- ▶ **r**: insertar un miembro (reemplazándolo si existe)

## Listar los archivos objetos de la biblioteca

```
> ar t libsaludo.a  
hola.o  
adios.o
```

# Construcción de bibliotecas – Estática

## Compilar

```
> gcc -Wall -c hola.c  
> gcc -Wall -c adios.c
```

## Combinar los archivos y generar la biblioteca

```
> ar cr libsaludo.a hola.o adios.o
```

- ▶ **c**: crear el archivo (\*.a)
- ▶ **r**: insertar un miembro (reemplazándolo si existe)

## Listar los archivos objetos de la biblioteca

```
> ar t libsaludo.a  
hola.o  
adios.o
```

(Ver comando `nm - man`)

# Construcción de bibliotecas – Dinámica

## Compilar

```
> gcc -Wall -c -fpic hola.c  
> gcc -Wall -c -fpic adios.c
```

# Construcción de bibliotecas – Dinámica

## Compilar

```
> gcc -Wall -c -fpic hola.c  
> gcc -Wall -c -fpic adios.c
```

- ▶ **-fpic**: genera código independiente de la posición (position independent code)



# Construcción de bibliotecas – Dinámica

## Compilar

```
> gcc -Wall -c -fpic hola.c  
> gcc -Wall -c -fpic adios.c
```

- ▶ **-fpic**: genera código independiente de la posición (position independent code)

## Combinar los archivos y generar la biblioteca

```
> gcc -shared hola.o adios.o -o libsaludo.so
```

