



## Módulos y bibliotecas

PPROG

## Índice de contenidos

- Módulo vs. librería
- Archivos de cabecera vs. archivos de implementación
- Compilar módulos y librerías

## Módulo vs. librería

### □ **Módulo**

- ▣ Cada uno de los bloques funcionales en los que se divide un programa
- ▣ Concepto procedente de la “*programación modular*”
- ▣ Ej. Cada uno de los módulos de vuestro proyecto.
  - space, object, game, etc.

### □ **Librerías o biblioteca**

- ▣ Rutinas de uso común que implementan operaciones comunes
- ▣ Ej.
  - Librería estándar de C → libc
  - Librería matemática → libm
  - Librería para gestión de terminal → libncurses
  - etc.

## Archivos de cabecera vs. archivos de implementación

### □ **Archivos de cabecera → .h**

- ▣ Definición de macros públicas
- ▣ Definición de tipos y estructuras públicos
- ▣ Declaraciones de variables públicas
- ▣ Declaración de funciones públicas (prototipos) → Interfaz

### □ **Archivos de implementación → .c**

- ▣ Declaraciones y definiciones privadas (macros, variables, tipos estructuras y funciones)
- ▣ Definición de funciones públicas → implementación

## Inclusión de cabeceras

### □ Cabeceras de sistema

- ▣ `#include <cabecera.h>`
- ▣ Se encuentran en directorios como `/usr/include`, `/usr/local/include`, etc.
  - Depende del sistema
- ▣ El compilador sabe encontrarlas para incluirlas en fase de preprocesado

### □ Cabeceras de nuestros desarrollos

- ▣ `#include "cabecera.h"`
- ▣ El compilador busca primero en el directorio de trabajo
- ▣ Si no lo encuentra, seguirá buscando en alguno de los directorios especificados con la opción `"-I<dir>"`
  - Ej. `gcc -I./include_dir -o executable mod1.c mod2.c main_loop.c`

## Compilar módulos y librerías (I)

### □ Paso 1: Proceso de pre-compilación

#### ▣ Incluir el archivo de cabecera en nuestro código

- `#include <stdio.h>`
- `#include <math.h>`
- `#include <curses.h>`
- ...

#### ▣ Si procede, usar opción `-I` del compilador (ver transp. anterior)

## Compilar módulos y librerías (II)

### □ Paso 2: Proceso de enlazado (*linkado*)

#### ▣ Indicar al compilador con qué librería se debe hacer el enlace

##### ■ Opción `-l<librería>`

■ Si se usa la librería `lib<LIBRARY>` habrá que compilar con `-l<LIBRARY>`

■ Librería matemática → `libm` → opción `-lm`

■ Librería estándar de C → `libc` → opción `-lc`

■ Librería para gestión de terminal → `libncurses` → `-lncurses`

#### ▣ Indicar al compilador dónde está la librería

##### ■ Por defecto el compilador buscará en directorios estándar:

■ `/usr/lib`, `/usr/local/lib`, etc.

##### ■ Sino, opción `-L<directorio>`

## Proceso de compilación enlazado con GCC

