Introduction to C

Natalia Bidart

FaMAF - UNC

27 de Abril de 2011

Intro

Source code (código fuente)

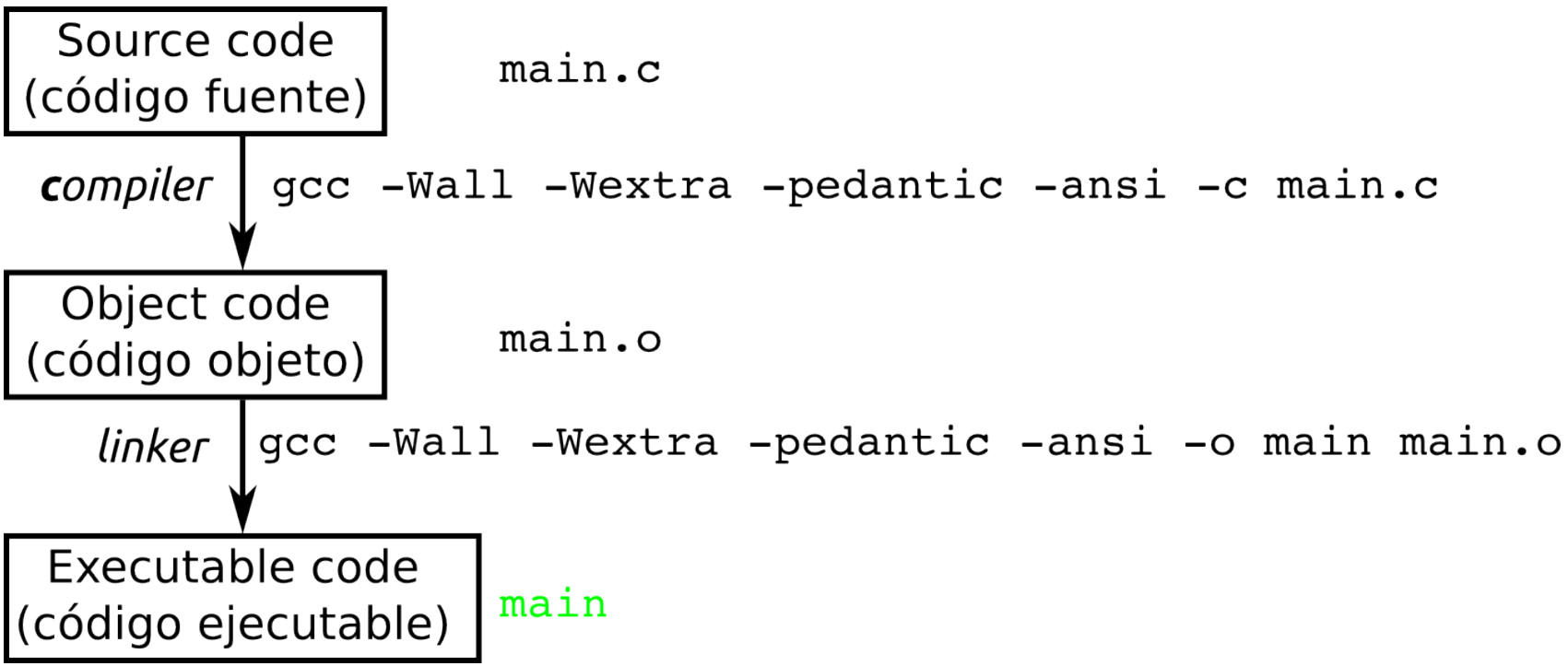
Compile

Link

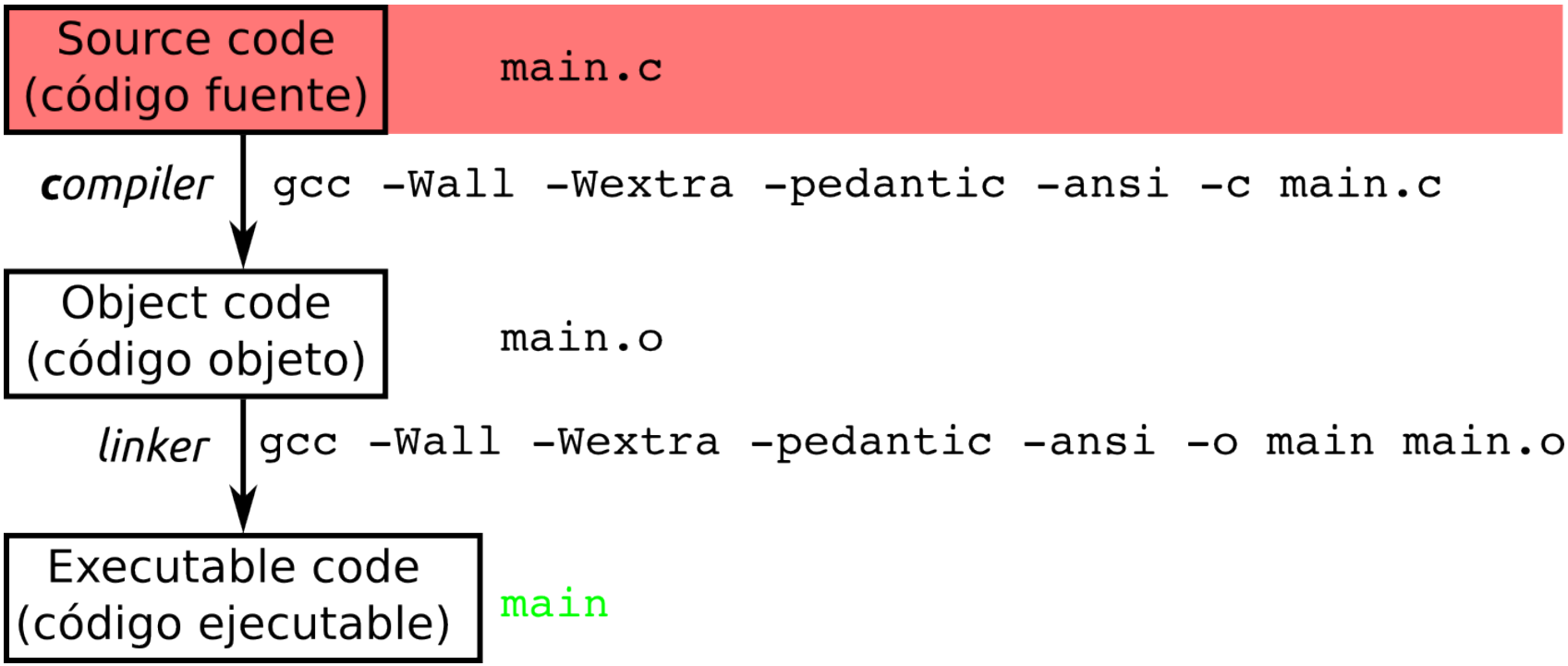
Execution

Specifics

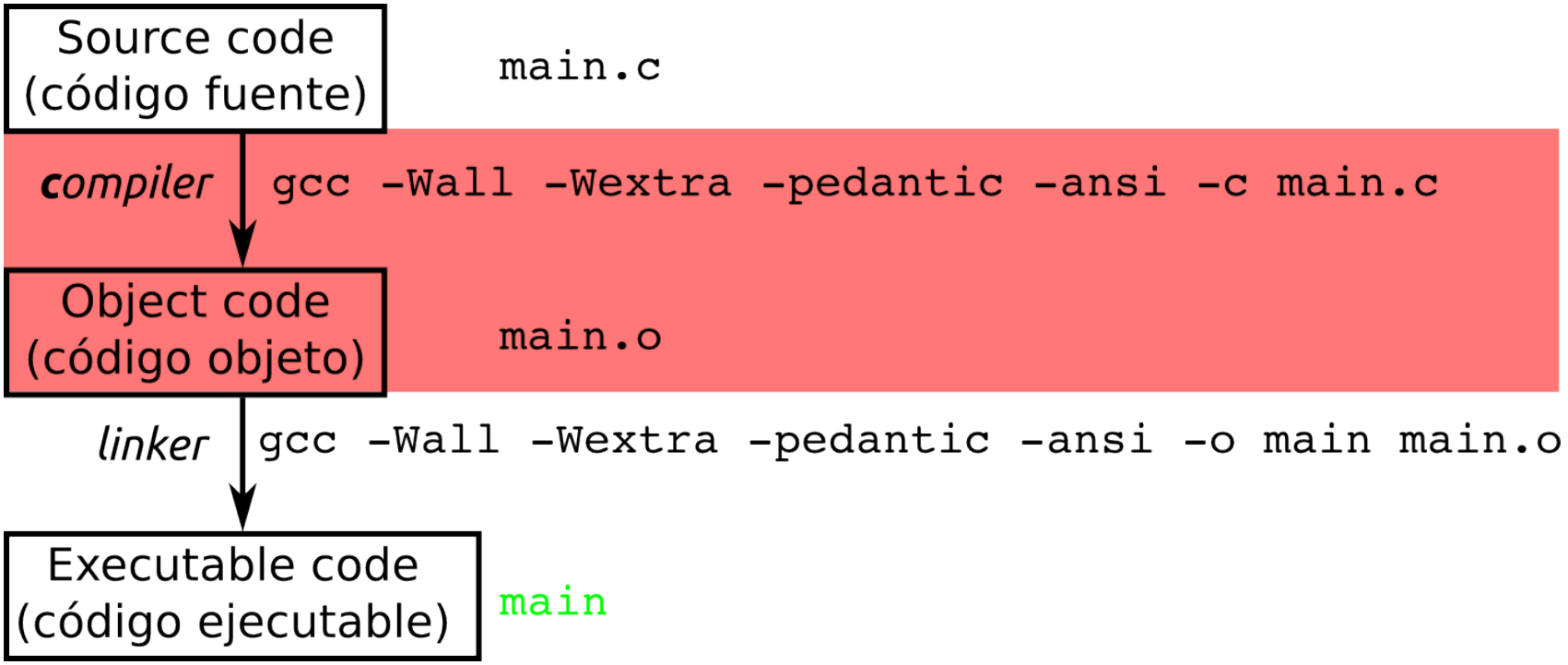
C code lifecycle (el ciclo de vida de código C)



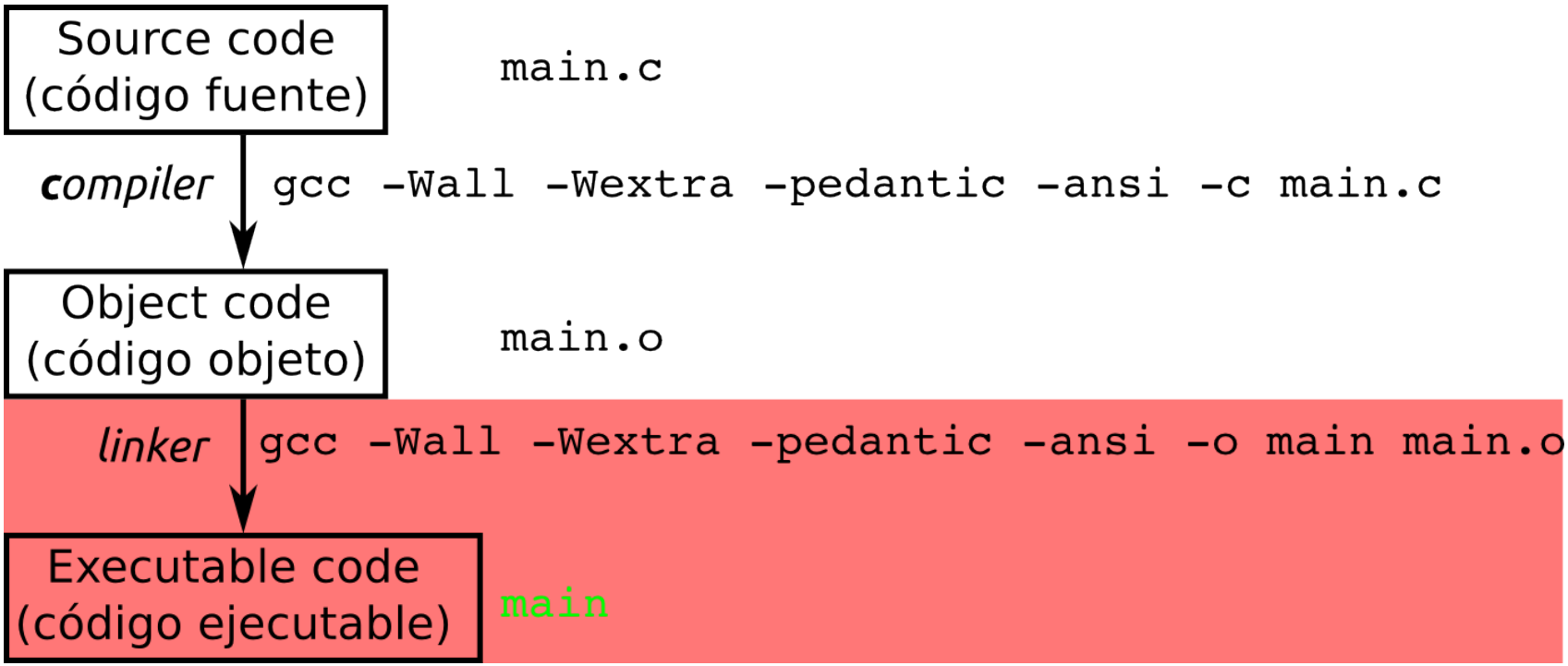
C code lifecycle (el ciclo de vida de código C)



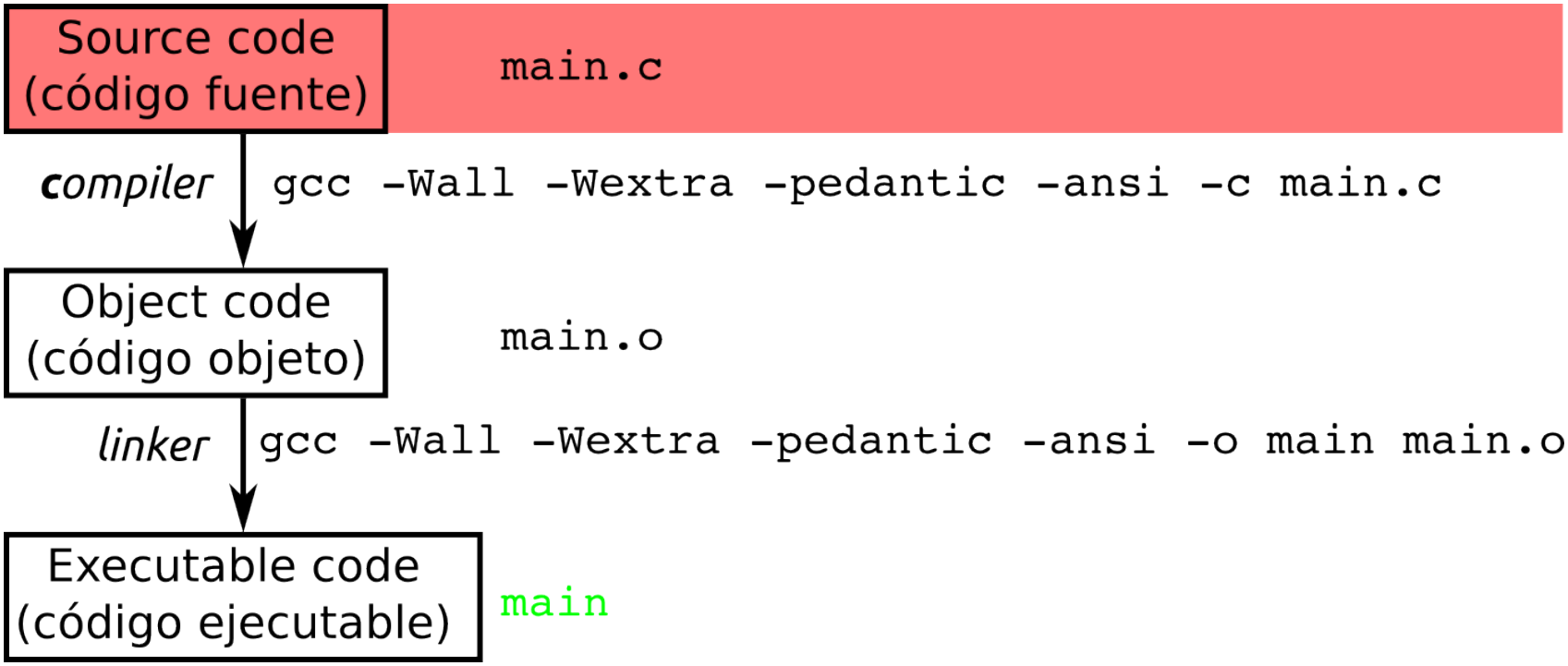
C code lifecycle (el ciclo de vida de código C)



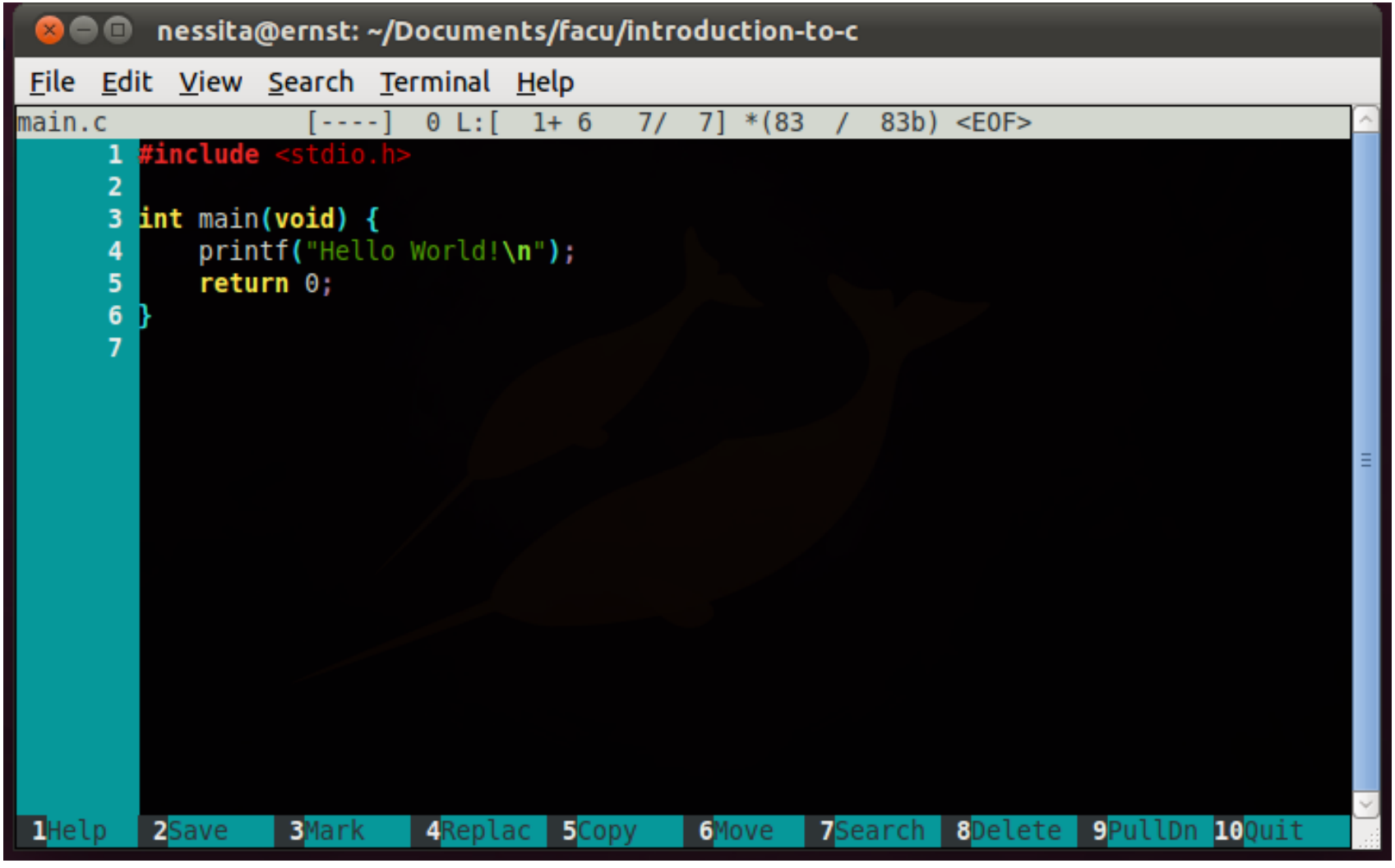
C code lifecycle (el ciclo de vida de código C)



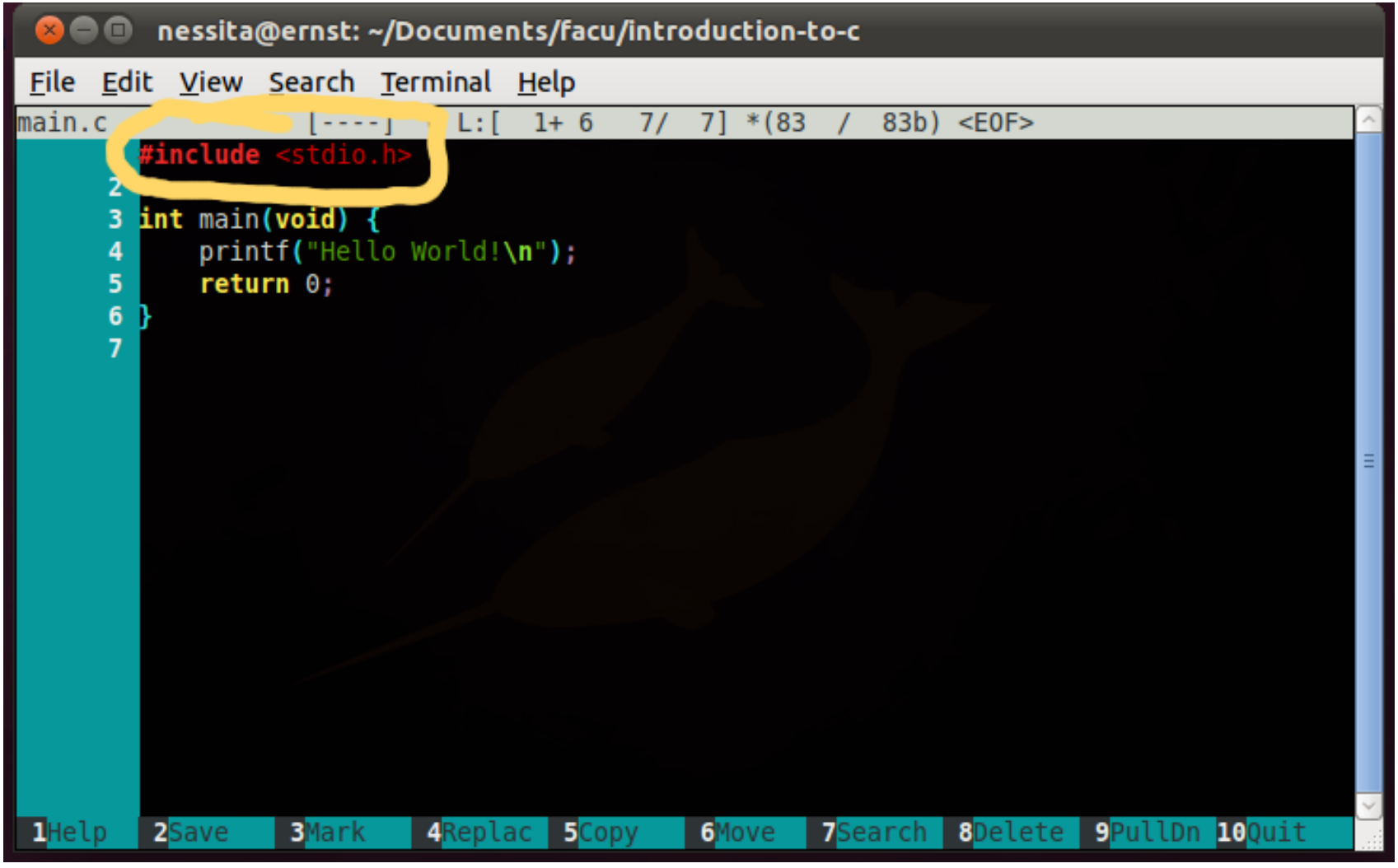
Source code (código fuente)



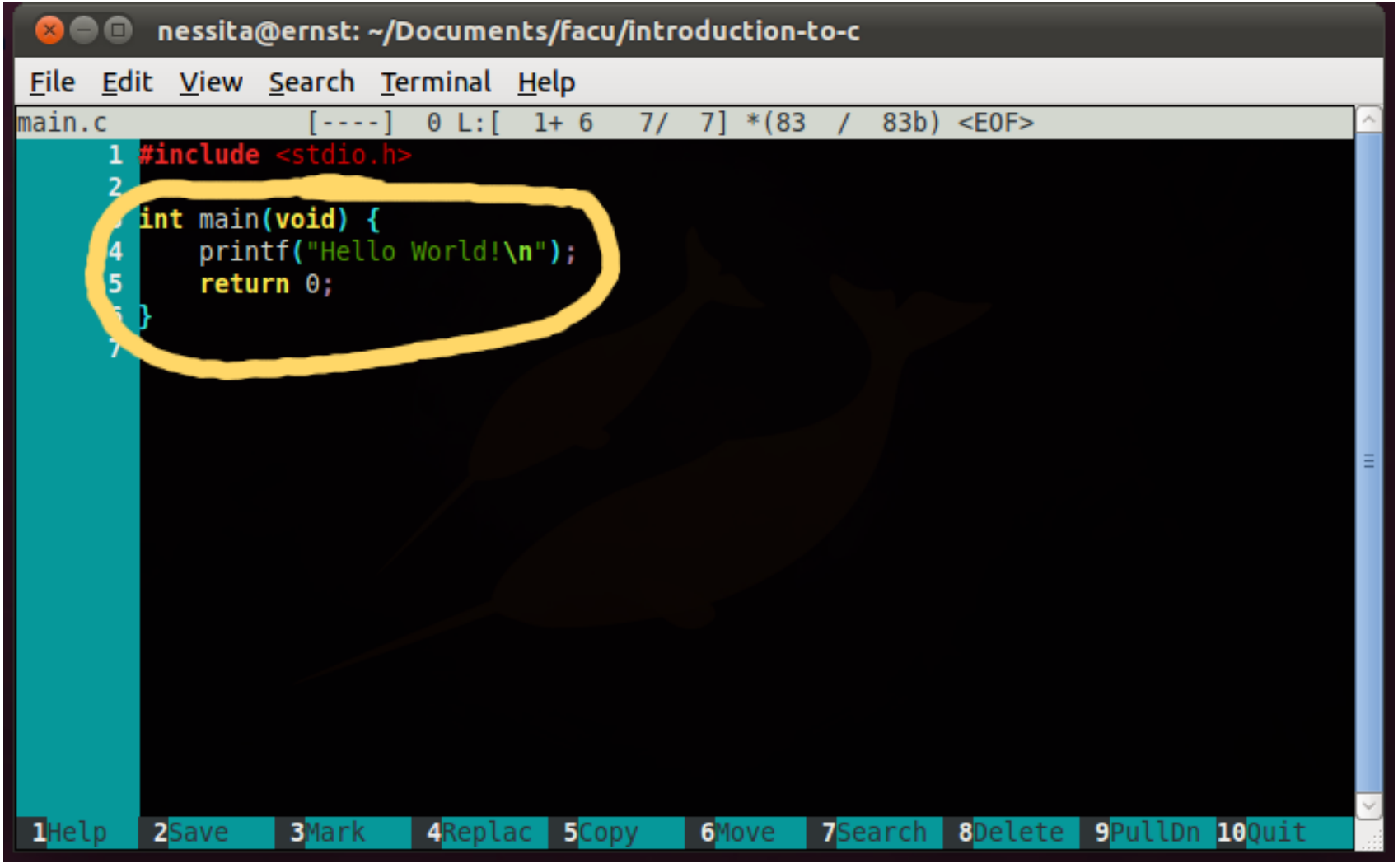
The simplest program ever
(el programa más simple de todos)



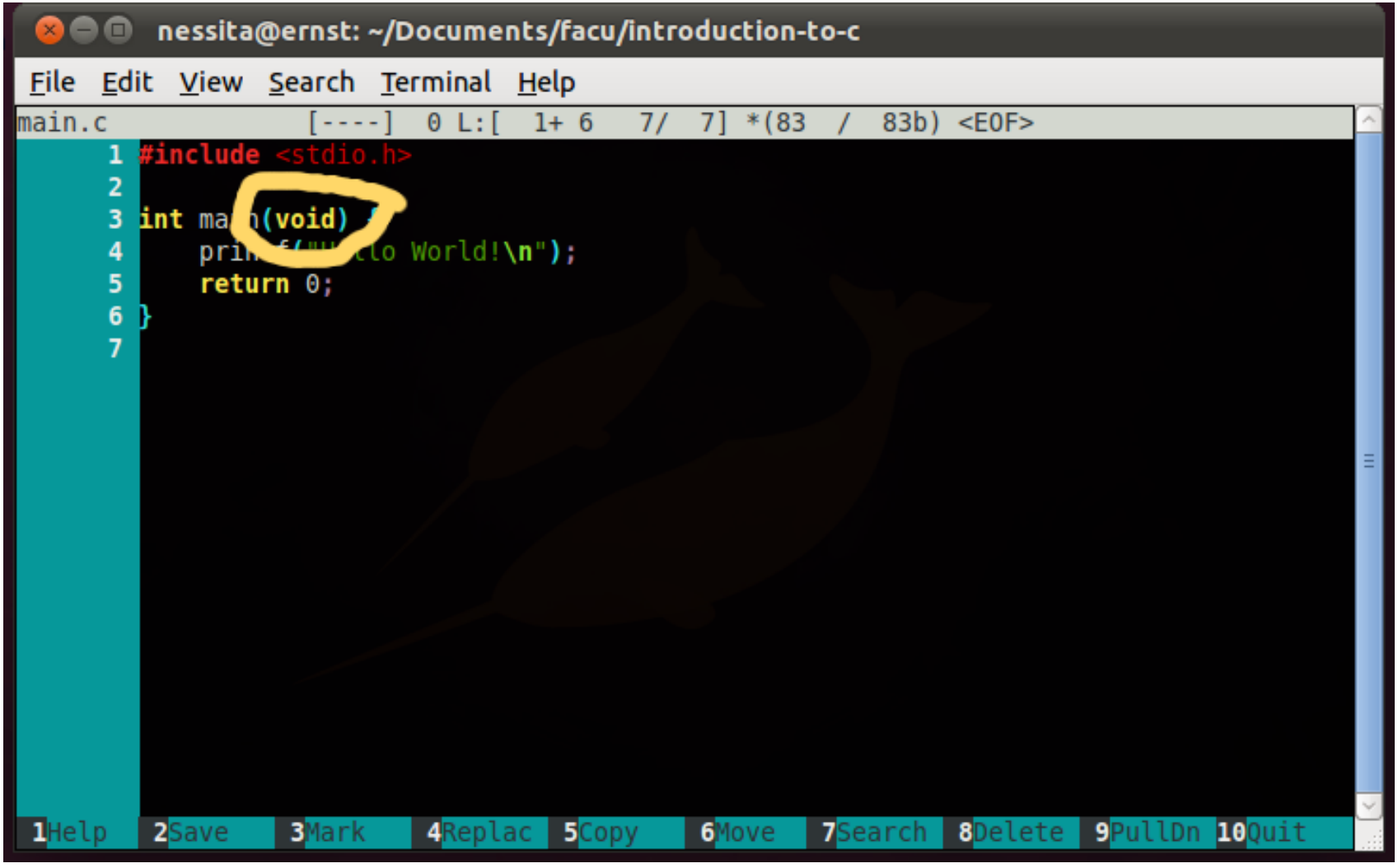
Including libraries
(incluyendo bibliotecas)



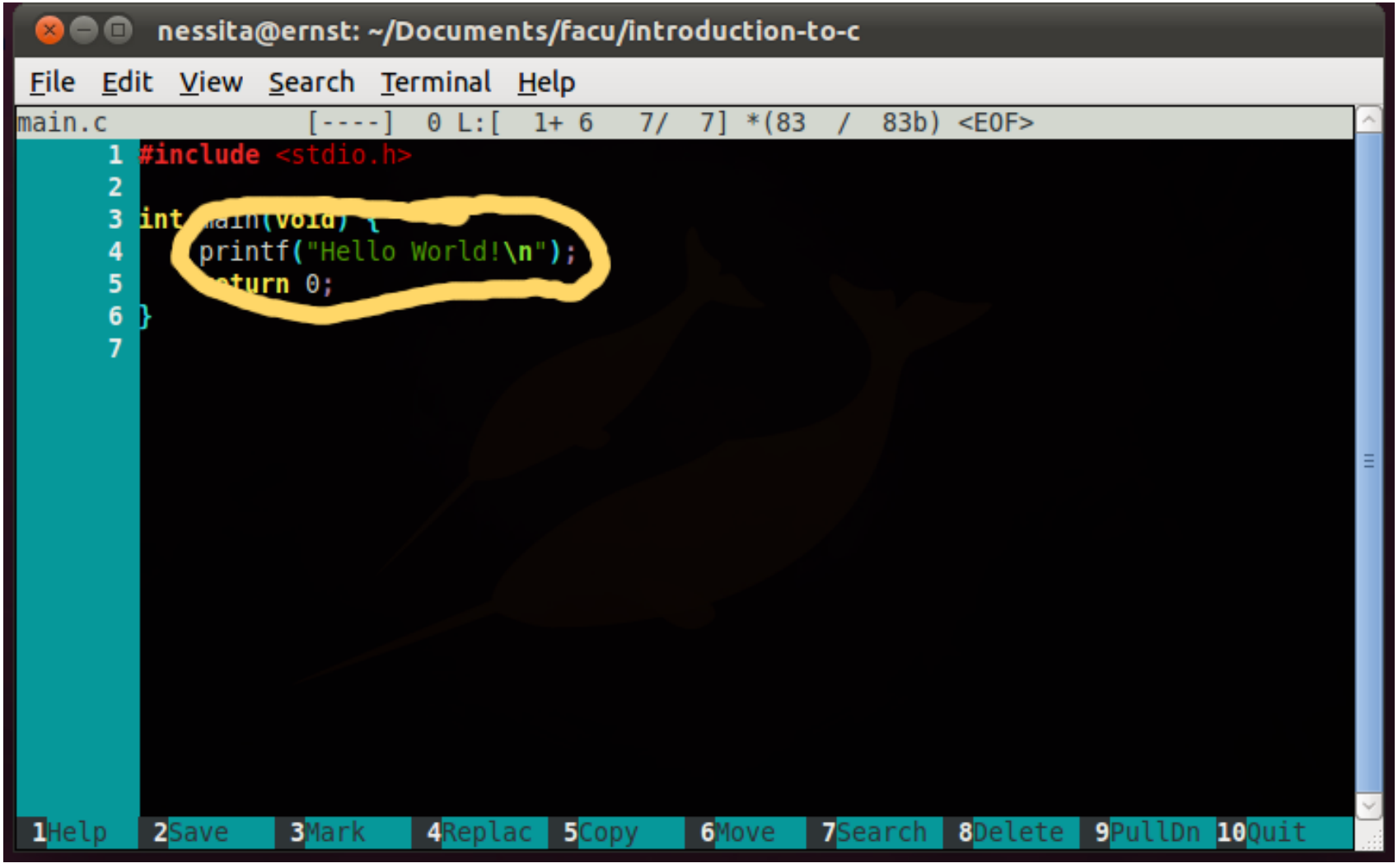
The entry point
(el punto de entrada)



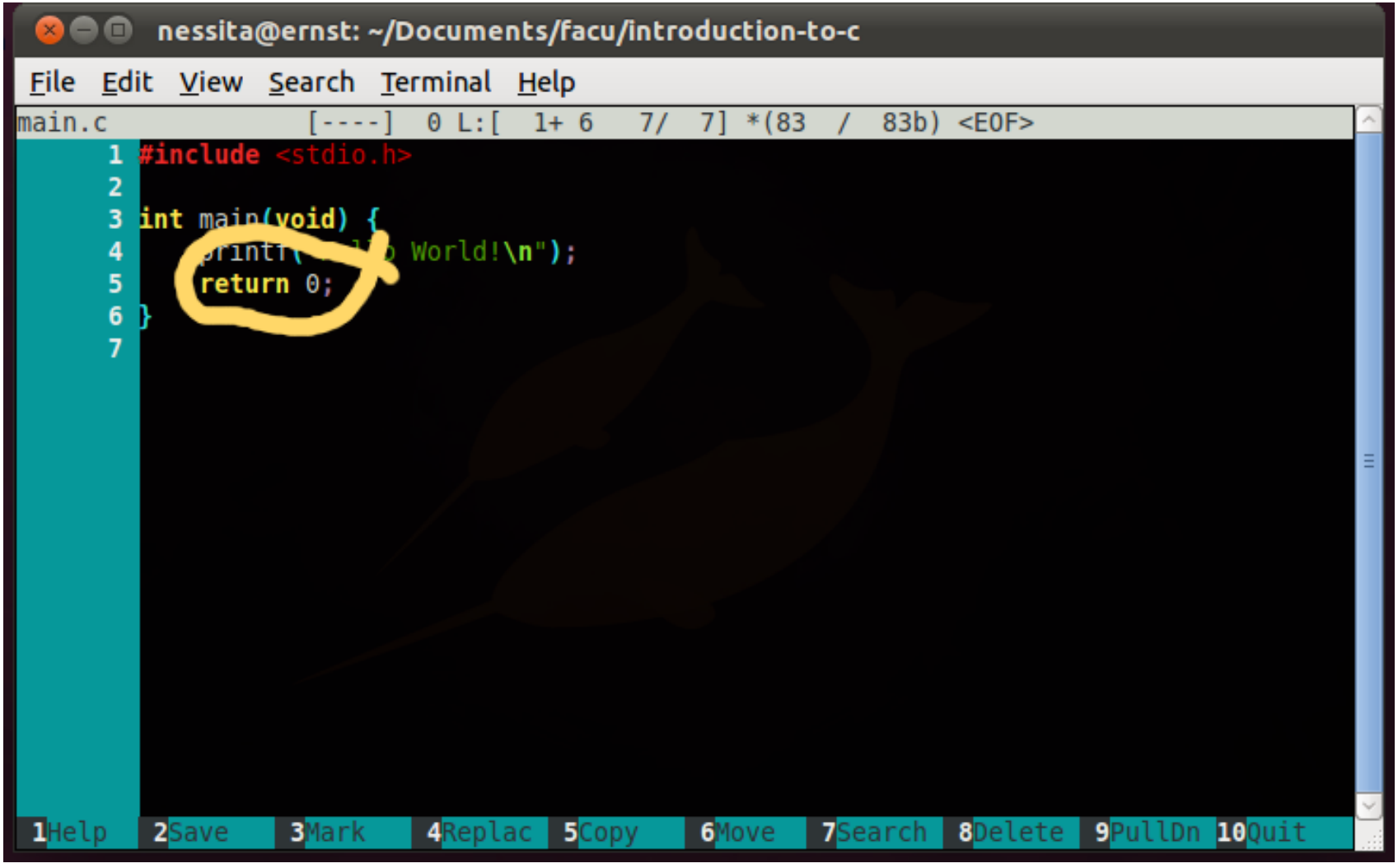
Explicitly saying no args
(explícitamente no aceptando argumentos)



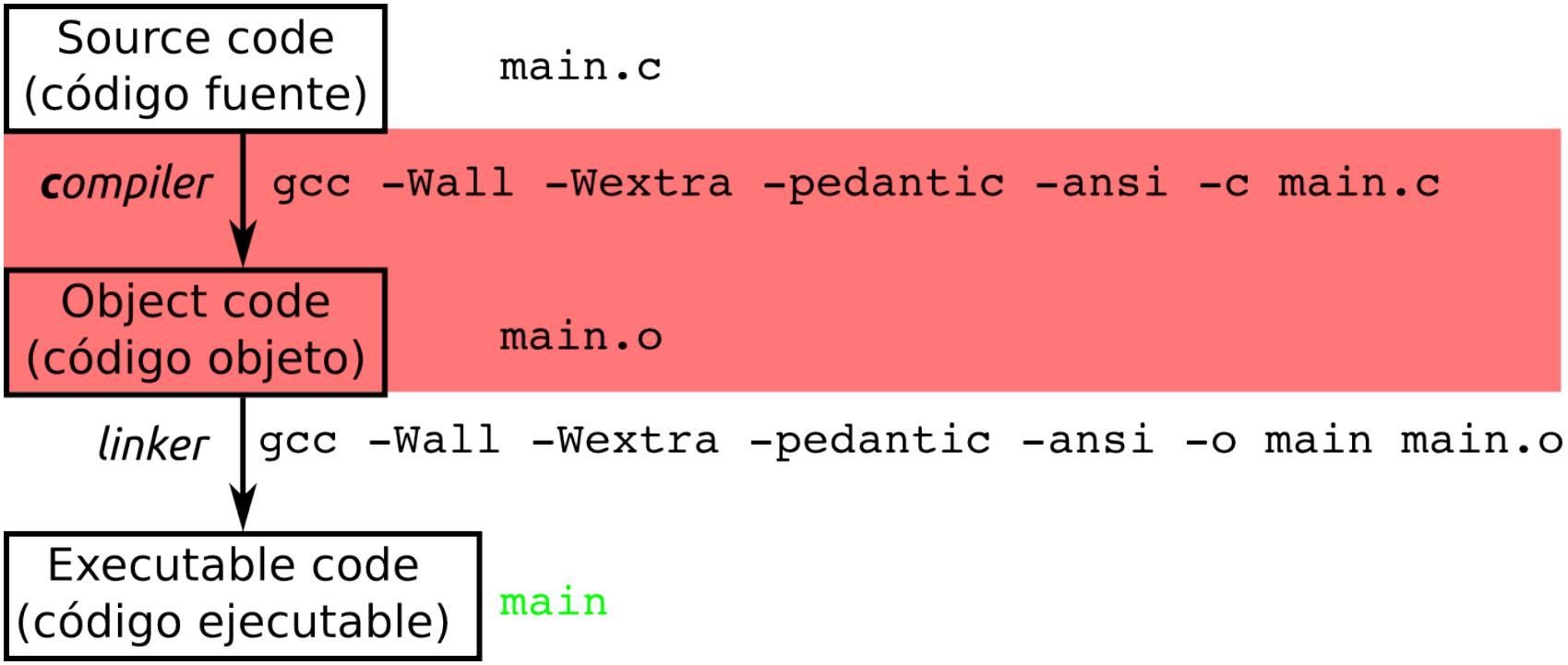
Interacting with the user
(interactuando con el usuario)



Returning values
(devolviendo valores como resultado)



Compile

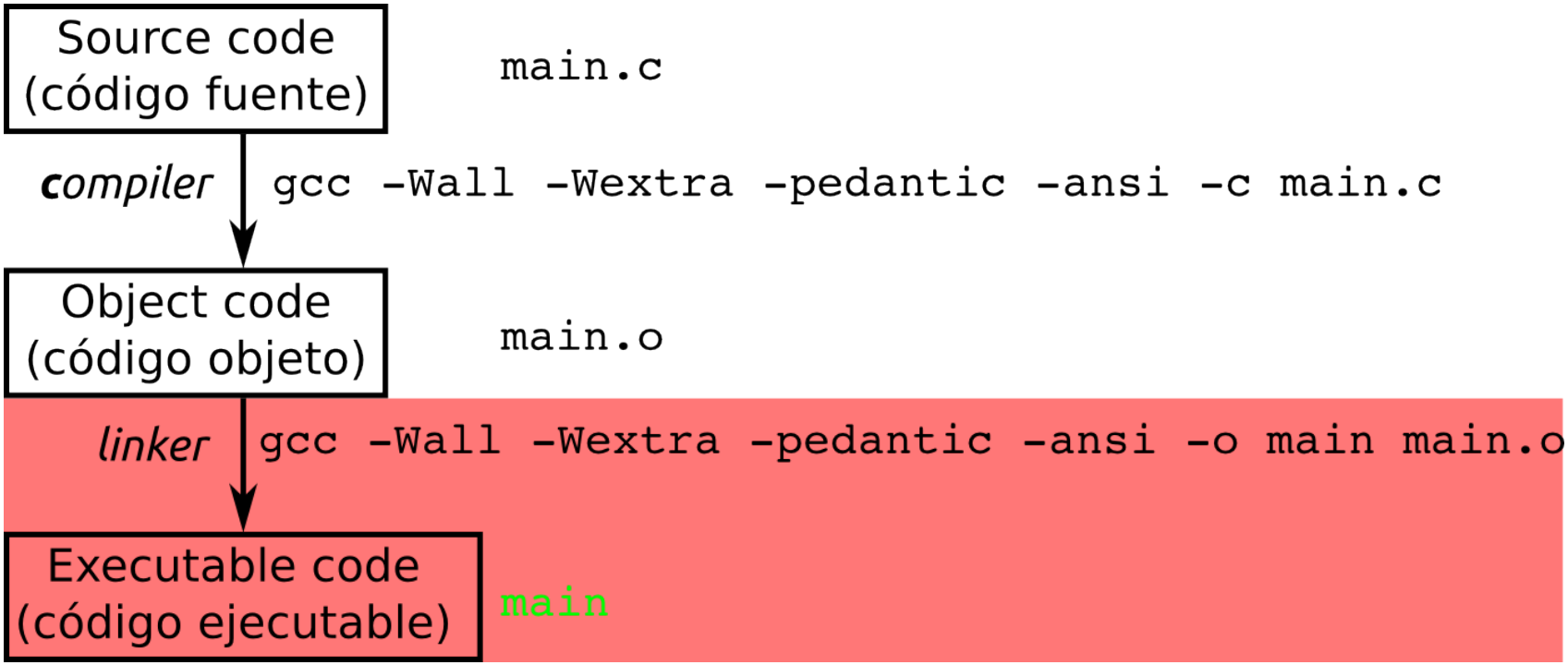

Source code
(código fuente)
main.c
compiler
gcc -Wall -Wextra -pedantic -ansi -c main.c
Object code
(código objeto)
main.o
linker
gcc -Wall -Wextra -pedantic -ansi -o main main.o
Executable code
(código ejecutable)
main

Compile

```
$ gcc -Wall -Wextra -pedantic -ansi -c main.c
```

gcc	The C compiler (el compilador de C)
-Wall	All the warnings (todas las advertencias)
-Wextra	Extra warnings (advertencias extras)
-pedantic	The strict warnings (advertencias estrictas)
-ansi	Standard C code (código C estándard)
-c	Compile (compilar)
main.c	One or more source code files (uno o más archivos de código fuente)

Link


Source code
(código fuente)
main.c
compiler
gcc -Wall -Wextra -pedantic -ansi -c main.c
Object code
(código objeto)
main.o
linker
gcc -Wall -Wextra -pedantic -ansi -o main main.o
Executable code
(código ejecutable)
main

Link

```
$ gcc -Wall -Wextra -pedantic -ansi -o main main.o
```

gcc	The C linker (el linker de C)
-Wall	All the warnings (todas las advertencias)
-Wextra	Extra warnings (advertencias extras)
-pedantic	The strict warnings (advertencias estrictas)
-ansi	Standard C code (código C estándard)
-o	Output executable filename, “a.out” by default (nombre del archivo de salida ejecutable, “a.out” por defecto)
main.o	One or more object code files (uno o más archivos de código objeto)

Execution (ejecución)

```
nessita@ernst:~/Documents/facu/introduction-to-c$ ./main
Hello World!
nessita@ernst:~/Documents/facu/introduction-to-c$
```

Variable declaration (declaración de variables)

```
<type> <variable_name>;
```

```
int i;
char c;
double f;
int *p;
```

- El estándard C90 require que las declaraciones de variables no se mezclen con código.
- Builtin types (los tipos del lenguaje):
  - int, unsigned int, long int, etc.
  - double
  - char
  - void
  - pointers (*)

Assignment (asignación)

=

```
int i, j;
char c;
i = 10;
j = 500;
c = 'a';
/* esto es un comentario */
```

- Se puede declarar variables y asignarles un valor al mismo tiempo (siempre y cuando el valor sea conocido) con:

```
int i = 10, j = 500;
char c = 'a';
```

Data Input (entrada de datos)

scanf

```
int i = 0;
convertions = scanf("%d", &i);
```

- stdin representa la entrada estándard, lo que se ingresa por teclado.
- El primer argumento es lo que llamamos el "especificador de conversión", que indica qué conversión hay que aplicarle a lo leído de la entrada stándard.
- El segundo argumento es un lugar en memoria para almacenar lo leido (ojo con los tamaños!).
- El resultado `convertions` indica cuántas conversiones exitosas se hicieron.

Data Output (salida de datos)

printf

```
int i = 0;
printed = printf("Lo leido es: %d\n", i);
```

- stdout representa la salida estándard, lo que se lee en la pantalla.
- El primer argumento es el texto a mostrar en la salida estándard, puede incluir especificadores de conversión (iguales que los que usa scanf).
- Éstos indican qué conversión hay que aplicarle a los parámetros subsiguientes.
- El resultado `printed` indica cuántos caracteres fueron exitosamente impresos en la pantalla.

Man pages (páginas de manual)

```
$ man scanf
$ man 3 printf
```

Paquetes necesarios:

- manpages-dev
- manpages-es (traducciones al español)
- glibc-doc (para `info`, que es otro comando para consultar manuales)

Operaciones básicas

- `+` Sum (suma)
- `-` Substraction (resta)
- `*` Multiplication (multiplicación)
- `/` Float division (división flotante)
- `%` Module (módulo)

Tarea

Escribir un programa que pida al usuario un entero, y muestre por pantalla el cuadrado del entero ingresado.