

Sesión 01: Contacto C

Programación para Sistemas

Ángel Herranz

2020-2021

Universidad Politécnica de Madrid

- *Transparencias con ejercicios* para hacer en clase
- *Hablad*¹ y compartir con vuestros compañeros. . .

¹La crisis de la COVID19 ha abierto muchas opciones.

- Transparencias con ejercicios para hacer en clase
- *Hablad*¹ y compartir con vuestros compañeros. . .

. . . *excepto* cuando Herranz explica

¹La crisis de la COVID19 ha abierto muchas opciones.

- Transparencias con ejercicios para hacer en clase
- *Hablad*¹ y compartir con vuestros compañeros. . .
 . . . *excepto* cuando Herranz explica
- Hojas de ejercicios para hacer durante la semana. . .

¹La crisis de la COVID19 ha abierto muchas opciones.

- Transparencias con ejercicios para hacer en clase
- *Hablad*¹ y compartir con vuestros compañeros. . .
 . . . *excepto* cuando Herranz explica
- Hojas de ejercicios para hacer durante la semana. . .
- o para empezarlas en clase si la clase va *demasiado lenta*

¹La crisis de la COVID19 ha abierto muchas opciones.

Hola mundo

- ¿Alguien ha escrito código C alguna vez?

²Un programa que escribe "¡Hola mundo!" en la pantalla (en la salida estándar, para ser más precisos).

Hola mundo

- ¿Alguien ha escrito código C alguna vez?
- ¿Qué haríais para escribir un “hola mundo”?²

²Un programa que escribe “¡Hola mundo!” en la pantalla (en la salida estándar, para ser más precisos).

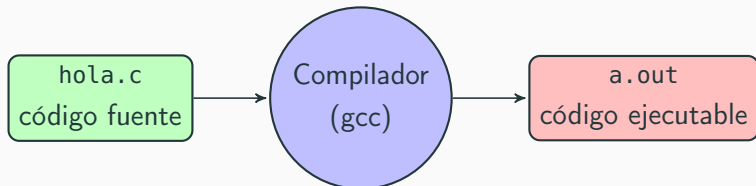


- ¿Alguien ha escrito código C alguna vez?
- ¿Qué haríais para escribir un “hola mundo”?²

²Un programa que escribe “¡Hola mundo!” en la pantalla (en la salida estándar, para ser más precisos).

Hola mundo

- ¿Alguien ha escrito código C alguna vez?
- ¿Qué haríais para escribir un “hola mundo”?²
- C es un lenguaje compilado



²Un programa que escribe "¡Hola mundo!" en la pantalla (en la salida estándar, para ser más precisos).

Leyenda (*be formal my friend*)

artefacto
(fichero)
desarrollado
por nosotros

artefacto
(fichero/memoria)
temporal

producto
(fichero)
final

artefacto
(fichero/biblioteca)
disponible
en el sistema

proceso
(programa en
ejecución)

¿Compilado? ¿Y cómo es...

- ...Java?
- ...Bash?
- ...Python?
- ...Haskell?
- ...Scala?
- ...Javascript?
- ...C#?

Necesitas un compilador de C

- Asumimos que vais a tener disponible una máquina Unix:
 1. Instala Linux en tu máquina
 2. Usa triqui (necesitas VPN y un cliente de ssh³ para conectar)
- Compilador de C:

GCC, the GNU Compiler Collection

- Tendrás que instalarlo en tu máquina (en Ubuntu lo mejor es instalar el paquete `build-essential`)
- Una vez instalado el compilador es `gcc` (normalmente programa `/usr/bin/gcc`)

³openssh disponible en Unix o PuTTY en Windows


Install party

🕒 20'

hola.c

```
#include <stdio.h>

int main()
{
    printf("¡Hola mundo!");
    return 0;
}
```

-  Crear un directorio `clases_pps`, en él un fichero `hola.c` con tu editor favorito, compilar y ejecutar.

hola.c

```
#include <stdio.h>

int main()
{
    printf("¡Hola mundo!");
    return 0;
}
```



Crear un directorio `clases_pps`, en él un fichero `hola.c` con tu editor favorito, compilar y ejecutar.

```
$ mkdir clases_pps
$ cd clases_pps
$ nano hola.c
$ gcc hola.c
$ ls -l
$ ./a.out
```

¿a.out?

- El nombre del ejecutable `a.out` es un `capricho` de los desarrolladores de gcc
- Vamos a darle un nombre más digno: `hola`
- ¿Cómo?
“Los programas tienen argumentos” “gcc es un programa”


“gcc tiene `argumentos`”

¿a.out?

- El nombre del ejecutable `a.out` es un **capricho** de los desarrolladores de gcc
- Vamos a darle un nombre más digno: `hola`
- ¿Cómo?

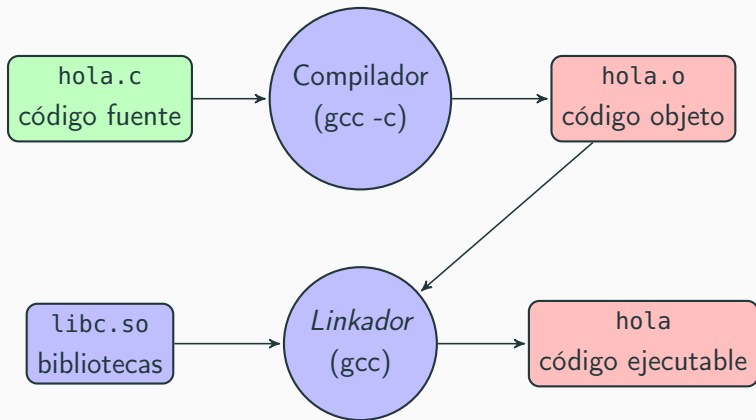
“Los programas tienen argumentos” “gcc es un programa”

“gcc tiene **argumentos**”

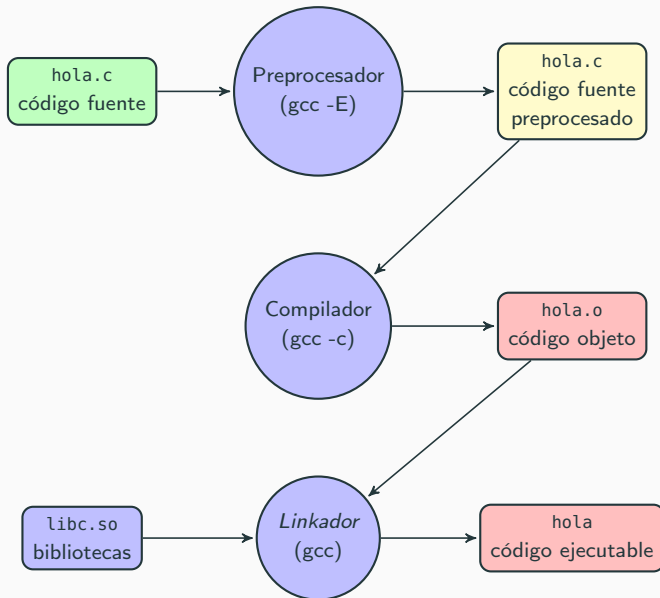
 gcc **-o hola** hola.c

-o file *Place output in file file.
This applies to whatever
sort of output is being
produced, [...] man gcc*

Proceso: compilación + *linkado*



Más detalle: preprocesado previo (**include**)






Hola ángel

- Comunicándonos con nuestro programa: argumentos desde la línea de comandos

Hola ángel

- Comunicándonos con nuestro programa: argumentos desde la línea de comandos

 Escribir un programa en C (hola.c) que saque por la salida estándar "¡Hola NOMBRE!", donde NOMBRE va a ser un argumento de la línea de comandos. Es decir, el programa tiene que hacer esto:

```
$ ./hola ángel  
¡Hola ángel!
```

Si el programa se invoca sin argumentos:

```
$ ./hola  
¡Hola mundo!
```

Hola ángel (solución)

Hola ángel (solución)

```
#include <stdio.h>
int main(int argc,
        char *argv[]) {
    char *quien;
    if (argc == 1) {
        quien = "mundo";
    }
    else {
        quien = argv[1];
    }
    printf("¡Hola %s\n",
        quien);
    return 0;
}
```

- **argc**: Números de argumentos con el que se invoca el programa desde la línea de comandos.
- argc siempre es mayor que 0 porque el nombre del programa se considera un argumento.
- **argv**: Argumentos del programa (*array de strings*).
- argv[0]: **Nombre** con el que se invoca el programa.
- argv[1]: Primer argumento.
- argv[2]: Segundo argumento.