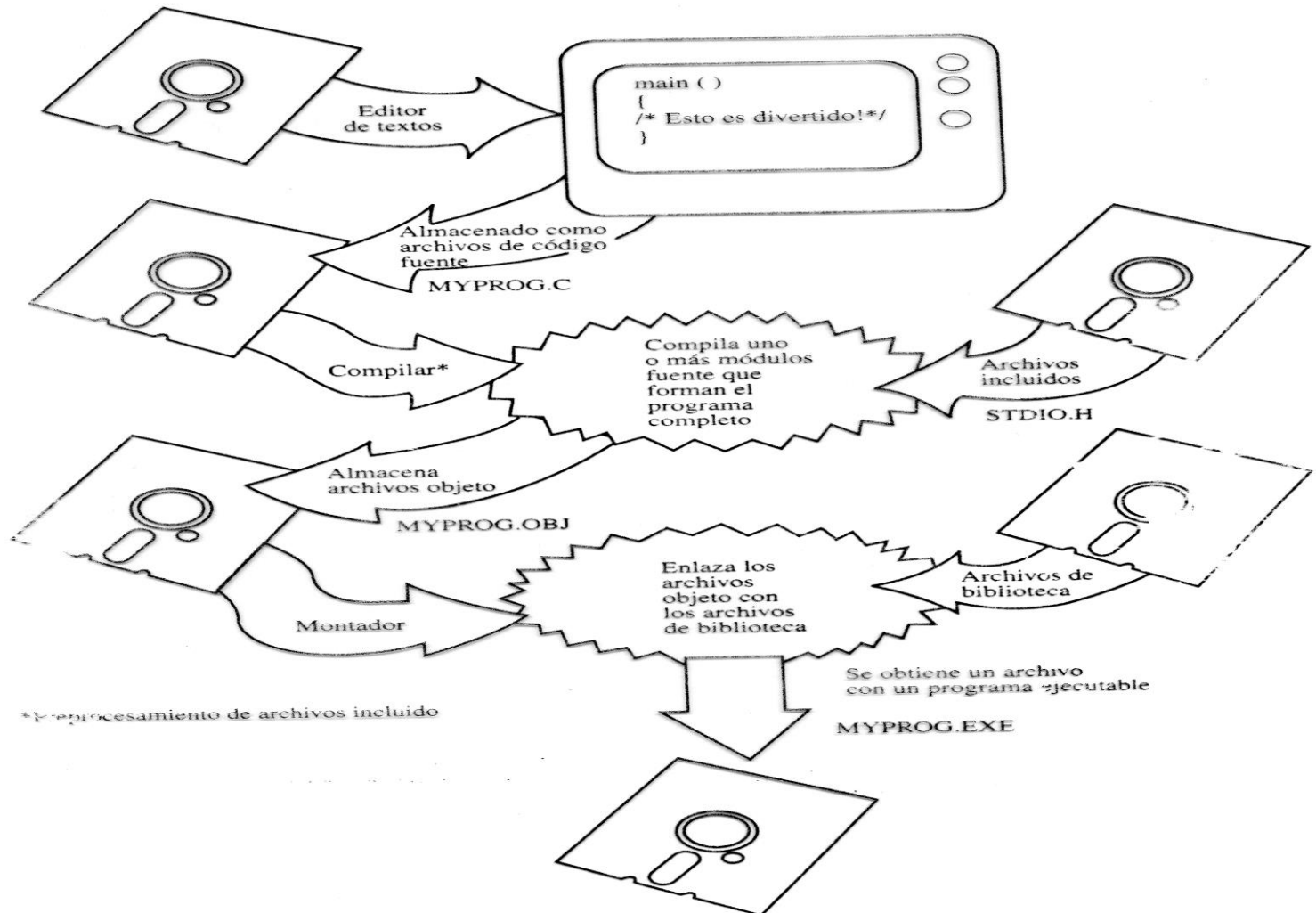


Editar, compilar, cargar, ejecutar...



El primer programa

```
/* Este es un programa C. Imprime un mensaje  
en la pantalla del computador */  
  
#include <stdio.h>  
  
int main()  
{  
    printf("¡Bienvenido a la programación en C! \n");  
    return 0;  
}
```

El primer programa

```
/* Este es un programa C. Imprime un mensaje  
en la pantalla del computador */
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    printf("¡Bienvenido a la programación en C! \n");
```

```
    return 0;
```

```
}
```

- Esto es un **comentario**
 - Comienza con `/*` y termina con `*/`
 - Puede ir en cualquier parte salvo entre comillas dobles:

```
"/*Esto no es un comentario sino una cadena de caracteres*/"
```

El primer programa

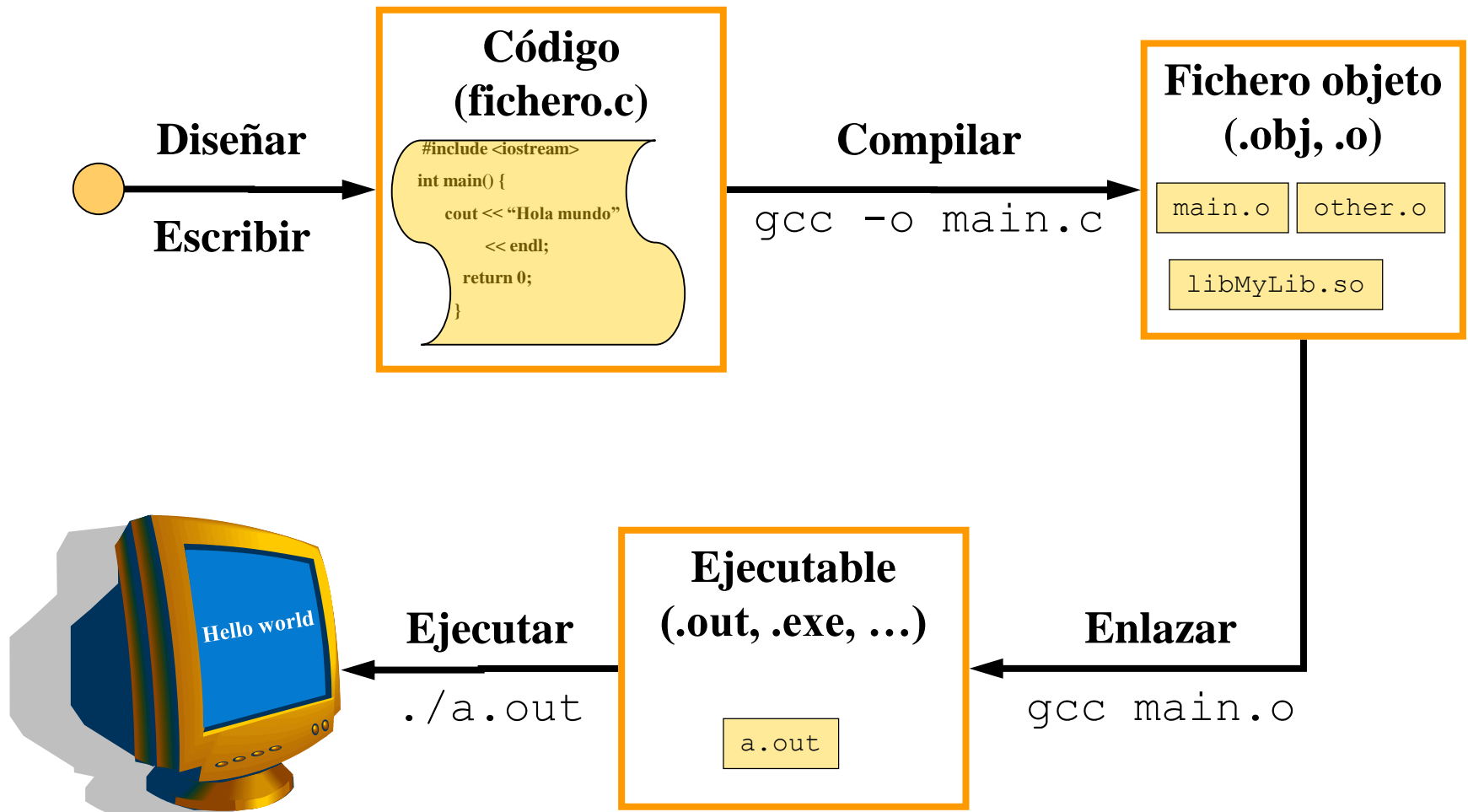
```
/* Este es un programa C. Imprime un mensaje  
en la pantalla del computador */
```

```
#include <stdio.h>
```

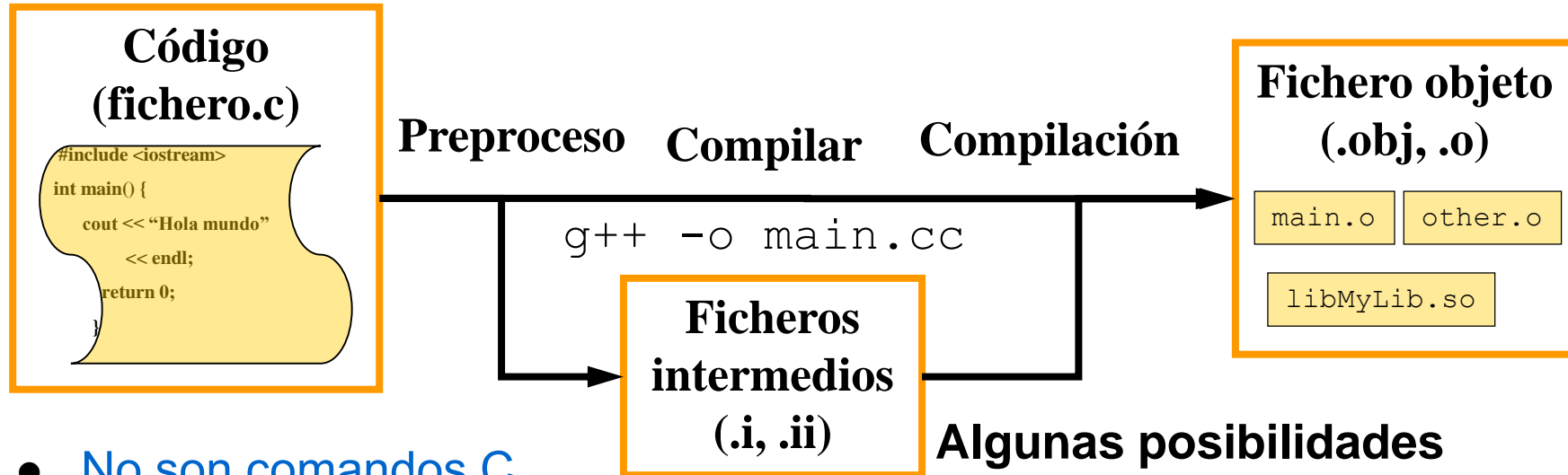
```
int main()  
{  
    printf("¡Bienvenido a la programación en C! \n");  
    return 0;  
}
```

- Esto es una directiva de preprocesador
- En este caso incluye el fichero que define las funciones de entrada/salida estándares

Escribir, compilar, enlazar, ejecutar



Directivas de preprocesador



- **No son comandos C...**
 - ... pero están presentes en cualquier compilador
- Muy utilizadas en C para:
- Se escriben con el símbolo `#` al comienzo
- Es una proceso intermedio antes de la propia compilación

Algunas posibilidades

- **Incluir código** de otros ficheros
- **Reemplazar** variables o símbolos
- **Definir constantes**
- Incluye cierta lógica

Directivas de preprocesador

- **#include** "header.h"
#include <header.h>
 - Incluye el fichero header.h
 - La 2ª forma se suele usar para “ficheros estándar”
 - ... algo subjetivo
- **#define** VAR value
 - Reemplaza la secuencia VAR por value en cualquier parte del código
- **#if, #ifdef, #ifndef, #else, #endif**
 - Permite activar o desactivar partes del código
 - A partir de variables de preprocesador
- **#pragma**
 - Mensajes al compilador. Poco utilizado

El primer programa

```
/* Este es un programa C. Imprime un mensaje  
en la pantalla del computador */  
  
#include <stdio.h>  
  
int main()  
{  
    printf("¡Bienvenido a la programación en C! \n");  
    return 0;  
}
```

- **main()** es el nombre de la **función principal** del programa
 - Es donde comienza la ejecución
 - Es obligatorio en todos los programas
- **'int'** indica que esta función devolverá un entero al acabar
 - Por convenio 0 indicaría que todo ha ido bien
 - Cualquier otro número indicaría que ha habido algún problema

El primer programa

```
/* Este es un programa C. Imprime un mensaje  
en la pantalla del computador */  
  
#include <stdio.h>  
  
int main()  
{  
    printf("¡Bienvenido a la programación en C! \n");  
    return 0;  
}
```

- Las **llaves** indican el **principio** y el **final** de un determinado **bloque de instrucciones**
 - En este caso indican el principio y el final de las instrucciones correspondientes a la función **main()**

El primer programa

```
/* Este es un programa C. Imprime un mensaje  
en la pantalla del computador */  
  
#include <stdio.h>  
  
int main()  
{  
    printf("¡Bienvenido a la programación en C! \n");  
    return 0;  
}
```

- **printf** es una función estándar para **imprimir** un texto por pantalla
 - Obsérvad que termina en ;

El primer programa

```
/* Este es un programa C. Imprime un mensaje  
en la pantalla del computador */  
  
#include <stdio.h>  
  
int main()  
{  
    printf("¡Bienvenido a la programación en C! \n");  
    return 0;  
}
```

- Esta instrucción **finaliza** (sale de) la función main() devolviendo 0
 - Que indicaría que todo ha ido bien
 - **return** es una de las palabras reservadas de C

Compilando, enlazando y ejecutando

```
[host] gedit ejercicio1.c
```

Abrimos un editor para escribir el código fuente y lo guardamos

```
.....
```

```
[host] ls
```

```
ejercicio1.c
```

```
[host] gcc ejercicio1.c
```

Compilar y enlazar. El resultado es...

```
[host] ls
```

```
ejercicio1.c
```

a.out

```
[host] ./a.out
```

Ejecutar

```
Bienvenido a la programacion en C!
```

Compilar y enlazar eligiendo el nombre del fichero de salida

```
[host] gcc ejercicio1.c -o ejercicio1.exe
```

```
[host] ls
```

```
a.out      ejercicio1.c
```

ejercicio1.exe

```
[host] ./ejercicio1.exe
```

Ejecutar

```
Bienvenido a la programacion en C!
```

Salida por pantalla

- La forma principal de escribir mensajes por pantalla es con `printf()`
- Para utilizarlo hay que incluir `stdio.h`
- En su versión más simple toma como argumentos un texto entre comillas
`printf("Algun texto");`
- No produce de manera automática un salto de línea → Para ello hay que añadir `\n`
`printf("Algun texto\n");`

Caracteres especiales

- La barra inversa (\) se utiliza para indicar algunos caracteres especiales
 - Para imprimir el símbolo de comillas ("): \"
 - Para imprimir la propia barra inversa: \\
 - Salto de línea: \n
 - Una tabulación: \t
 - Un pitido (*beep*): \a
 - Borrado: \b

```
#include <stdio.h>

int main() {
    printf("Aquí no hay un salto de línea...");
    printf("Pero aquí sí \n");
    printf("Esto lo digo \"entre comillas\" \n");
    printf("Aquí pongo una barra inversa: \\ \n");

    return 0;
}
```