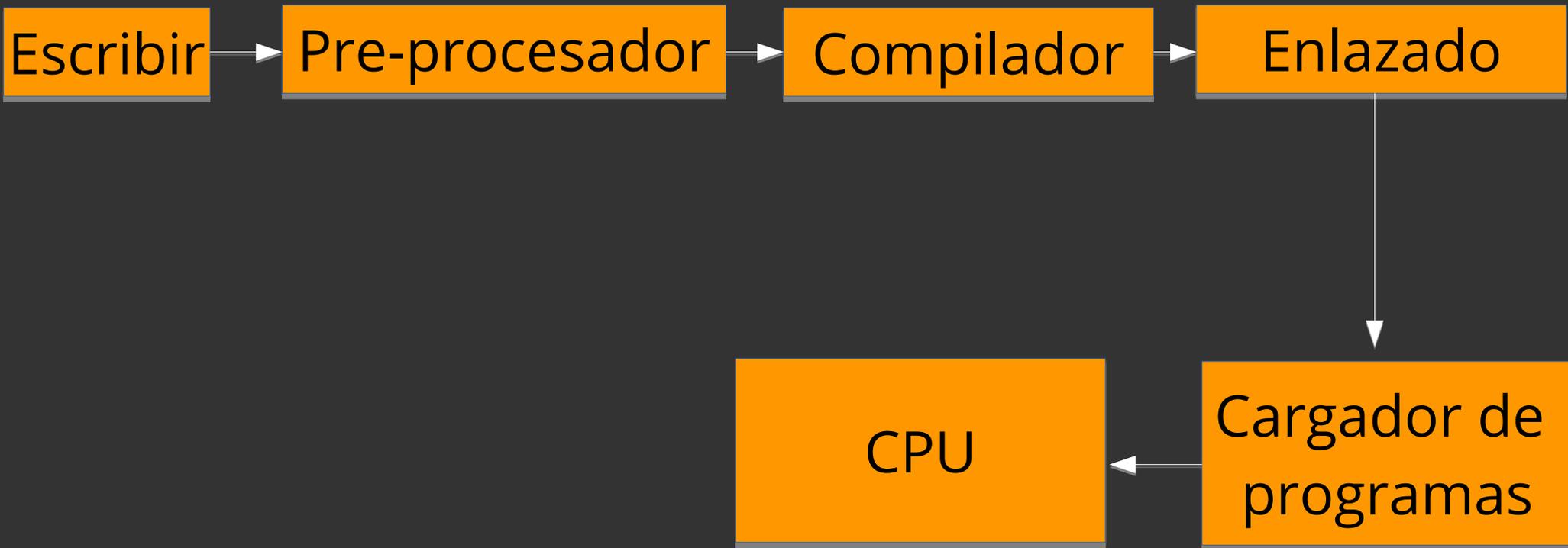


Intro a GCC

Nievas Martin

26/03/20

Pasos típicos para desarrollar un programa en C



Escribir

Pre

Comp

Enlazado

Carga

Ejecutar

Editores

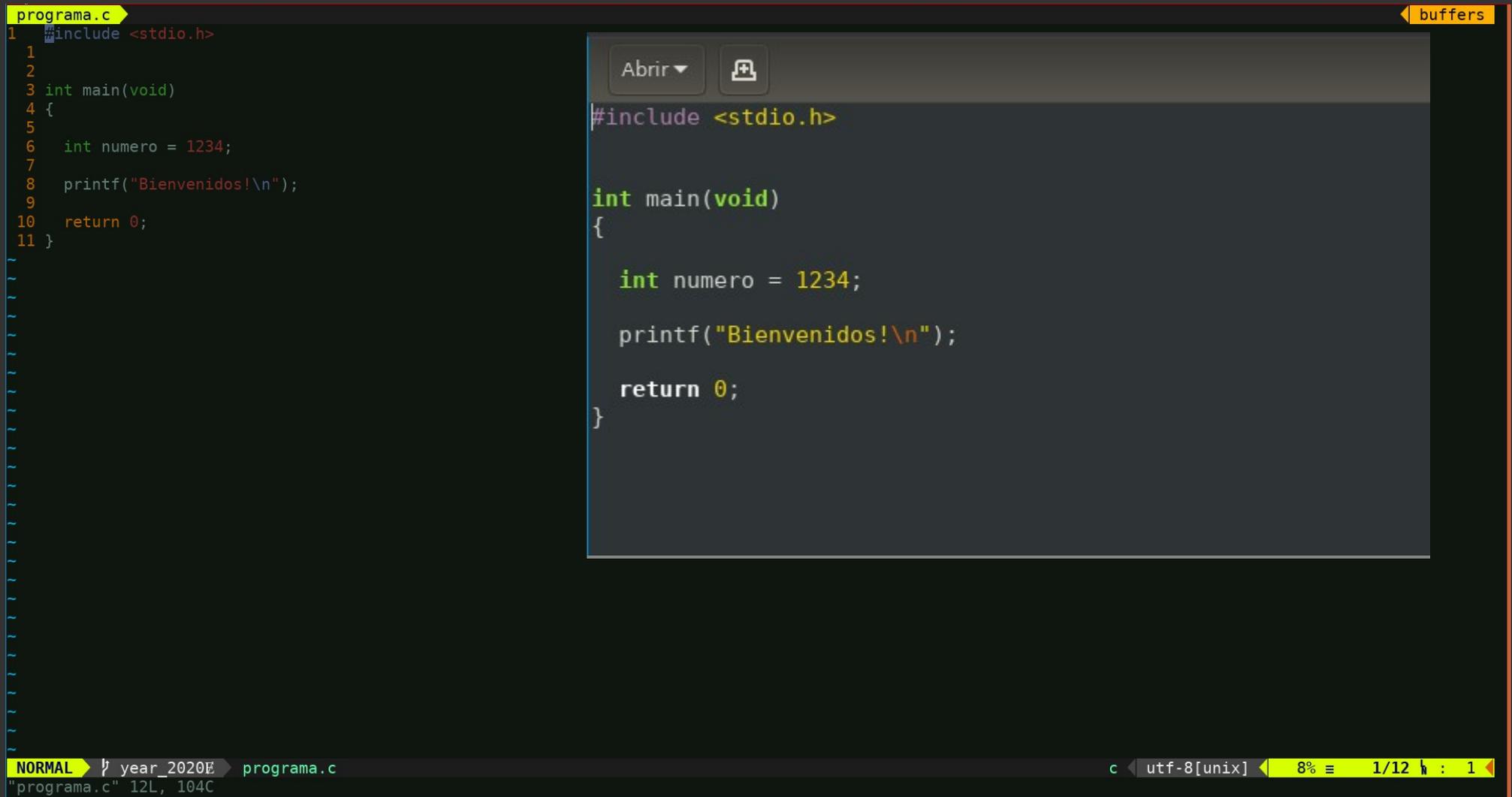
Escribir

Primero hay que escribir el programa (o descargarlo)

Podemos utilizar:

- Editor de texto

Editores – VIM, gedit, notepad, ...



The image shows a VIM editor window with a dark background. The main editor area displays the following C code:

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5
6     int numero = 1234;
7
8     printf("Bienvenidos!\n");
9
10    return 0;
11 }
```

On the right side of the editor, there is a floating window titled 'Abrire' with a plus icon. This window contains a zoomed-in view of the code from the main editor:

```
#include <stdio.h>

int main(void)
{

    int numero = 1234;

    printf("Bienvenidos!\n");

    return 0;
}
```

The status bar at the bottom of the VIM window shows: 'NORMAL' mode, cursor at line 1, column 1, filename 'programa.c', encoding 'utf-8[unix]', 8% zoom, and page 1 of 12.

GCC - pre procesador

```
graph LR; A[Escribir] --> B[Pre-procesador]; B --> C[ ];
```

Escribir

Pre-procesador

Ejecutar el pre-procesador.
El preprocesador de C no es parte del compilador, es un paso separado en el proceso de compilación. Puede considerarse como una herramienta de sustitución de texto y le indica al compilador que realice el preprocesamiento requerido antes de la compilación real.

GCC - pre procesador

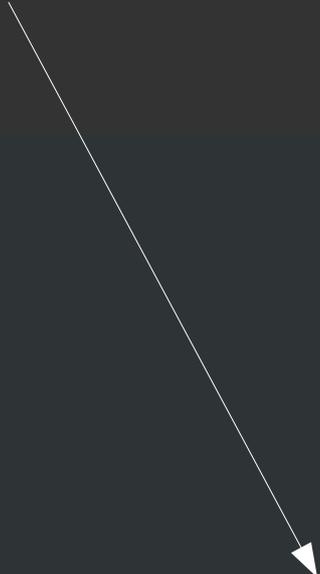
```
$ gcc -E programa.c > programa.i
```

```
extern void funlockfile (FILE *__stream) __a
# 868 "/usr/include/stdio.h" 3 4

# 2 "programa.c" 2

# 4 "programa.c"
int main(void)
{
    int numero = 1234;
    printf("Bienvenidos!\n");
    return 0;
}
```

Le agrega mucho texto al inicio del programa. Notar la cantidad de líneas del archivo programa.i



GCC - compilador



Un compilador es un programa que traduce código escrito en un lenguaje de programación a otro idioma.

GCC – compilar

```
$ gcc -c programa.i
```

```
$ gcc -c programa.c
```

GCC – compilar

```
$ gcc -c programa.i
```

```
$ gcc -c programa.c
```



El símbolo \$ significa que estamos en una terminal, para el caso de estos slides, una terminal de Linux

GCC – compilar

```
$ gcc -c programa.i
```

```
$ gcc -c programa.c
```

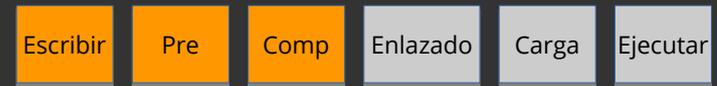
gcc puede compilar desde un .c también

Cualquiera de estos pasos nos genera un archivo:

programa.o

Éste ya no se puede leer por humanos (al menos no fácilmente)

GCC – compilar



```
$ gcc -c programa.i
```

```
$ gcc -c programa.c
```

Notar que de ésta forma, nos saltamos/ahorramos un paso

gcc puede compilar desde un .c también

Cualquiera de estos pasos nos genera un archivo:

programa.o

Éste ya no se puede leer por humanos (al menos no fácilmente)

GCC - enlazar



Los programas C generalmente contienen referencias a funciones definidas en otros lugares, como en las bibliotecas estándar o en las bibliotecas privadas de grupos de programadores que trabajan en un proyecto en particular. El código objeto producido por el compilador de C generalmente contiene “huecos” debido a estas partes faltantes. Un enlazador (o linker) vincula el código del objeto con el código de las funciones que faltan para producir una imagen ejecutable (sin piezas faltantes).

GCC - enlazar

Para nuestro ejemplo:

```
$ gcc programa.o -o programa.out
```

Pero podríamos haber requerido otra biblioteca:

```
$ gcc programa.o my_bib.o -o programa.out
```

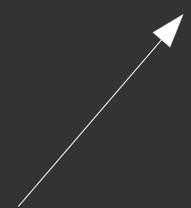
GCC - enlazar

Para nuestro ejemplo:

También podemos pasar del
.c al .out directamente

```
$ gcc programa.o -o programa.out
```

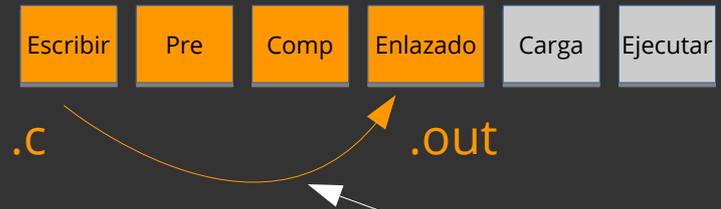
```
$ gcc programa.c -o programa.out
```



Pero podríamos haber requerido otra biblioteca:

```
$ gcc programa.o my_bib.o -o programa.out
```

GCC - enlazar



También podemos pasar del
`.c` al `.out` directamente

Para nuestro ejemplo:

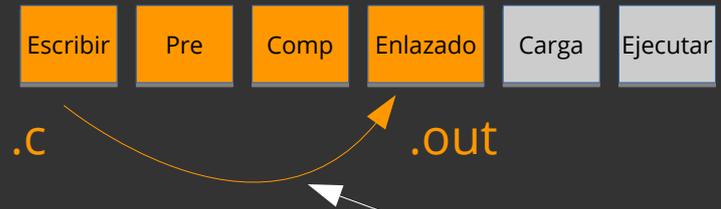
```
$ gcc programa.o -o programa.out
```

```
$ gcc programa.c -o programa.out
```

Pero podríamos haber requerido otra biblioteca:

```
$ gcc programa.o my_bib.o -o programa.out
```

GCC - enlazar



Para nuestro ejemplo:

```
$ gcc programa.o -o programa.out
```

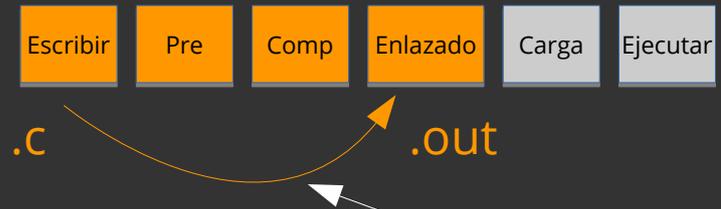
```
$ gcc programa.c -o programa.out
```

También podemos pasar del
.c al .out directamente
(en la práctica hacemos esto)

Pero podríamos haber requerido otra biblioteca:

```
$ gcc programa.o my_bib.o -o programa.out
```

GCC - enlazar



Para nuestro ejemplo:

```
$ gcc programa.o -o programa.out
```

```
$ gcc programa.c -o programa.out
```

También podemos pasar del
.c al .out directamente
(en la práctica hacemos esto)

Pero podríamos haber requerido otra biblioteca:

```
$ gcc programa.o -lm -o programa.out
```

Escribir

Pre

Comp

Enlazado

Carga

Ejecutar

Cargador de programas

Escribir

Pre-procesador

Compilador

Enlazado

Antes de que se pueda ejecutar un programa, primero debe ser colocado en la memoria. Esto lo hace el cargador, que toma la imagen ejecutable del disco (programa.out) y la transfiere a la memoria. También se cargan componentes adicionales de bibliotecas compartidas que utiliza el programa.

Cargador de programas

Escribir

Pre

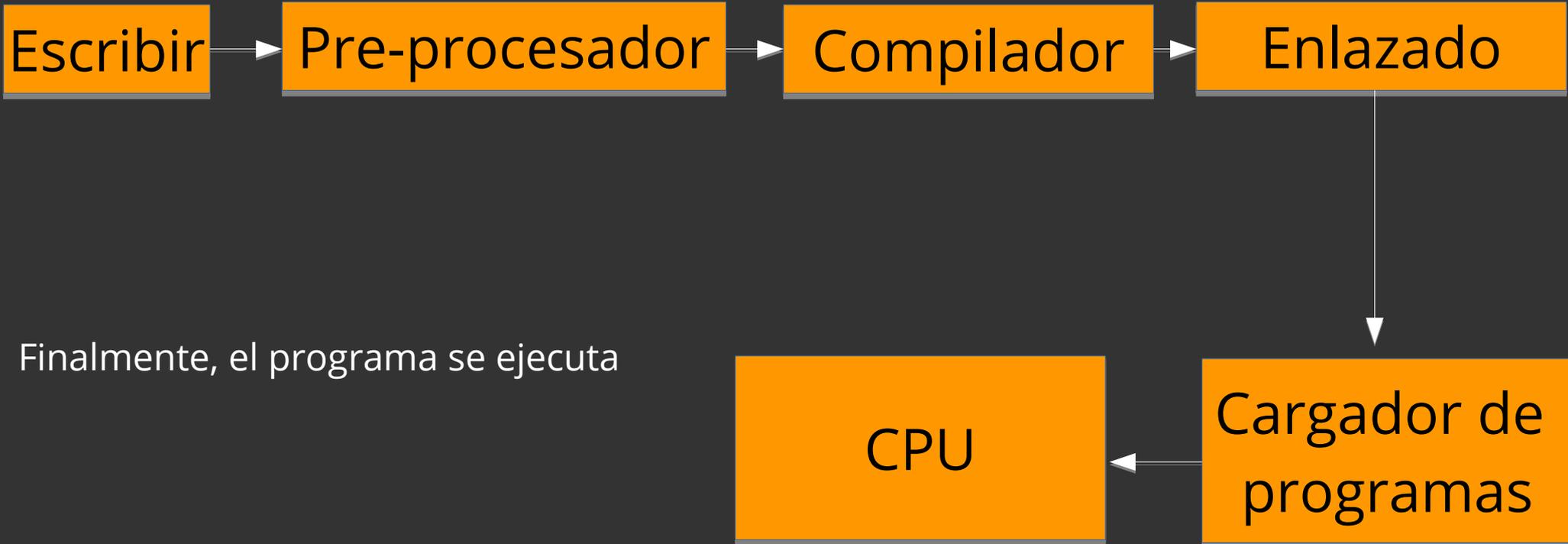
Comp

Enlazado

Carga

Ejecutar

Ejecutar el programa



Ejecutar el programa

En una terminal:

```
$ ./programa.out
```

Limpiar la pantalla:

```
$ clear
```

Repaso

Compilar desde el código fuente:

```
$ gcc programa.c -o programa.out
```

```
$ ./programa.out
```

GCC – pasos intermedios

```
$ gcc -save-temps programa.c
```

También podemos ver el assembler:

```
$ gcc -S programa.c
```

Consultas



mnievas@frc.utn.edu.ar

Edificio Salcedo Of. 5

Consultas