

5 Desarrollo y Carga de Programas de C en Memoria: Librerías estáticas y dinámicas

Introducción a los Sistemas Operativos,
2021-2022

Pablo González Nalda

Depto. de Lenguajes y Sistemas Informáticos
EU de Ingeniería de Vitoria-Gasteiz,
UPV/EHU



8 de marzo de 2022



Contenidos de la presentación

CONTENIDOS

Gestión de memoria

Fases de la creación del ejecutable

Reubicación

Librerías de enlace estático y dinámico

¿Más preguntas?

- 1 Gestión de memoria
- 2 Fases de la creación del ejecutable
- 3 Reubicación
- 4 Librerías de enlace estático y dinámico
- 5 ¿Más preguntas?



CONTENIDOS

Gestión de memoria

Introducción

Estructura de un programa en memoria, en ejecución

Fases de la creación del ejecutable

Reubicación

Librerías de enlace estático y dinámico

¿Más preguntas?

- 1 **Gestión de memoria**
- 2 Fases de la creación del ejecutable
- 3 Reubicación
- 4 Librerías de enlace estático y dinámico
- 5 ¿Más preguntas?



Introducción

CONTENIDOS

Gestión de memoria

Introducción

Estructura de un programa en memoria, en ejecución

Fases de la creación del ejecutable

Reubicación

Librerías de enlace estático y dinámico

¿Más preguntas?

La memoria es un recurso que debe regular el SO.

El SO debe proveer de métodos para que varios programas se ejecuten concurrentemente.



Estructura de un programa en memoria, en ejecución

CONTENIDOS

Gestión de memoria

Introducción

Estructura de un programa en memoria, en ejecución

Fases de la creación del ejecutable

Reubicación

Librerías de enlace estático y dinámico

¿Más preguntas?

Estructura de un programa en memoria, en ejecución:

- Datos estáticos
- Montículo, memoria dinámica
- Pila
- Instrucciones del programa
- Funciones de biblioteca (librerías)



CONTENIDOS

Gestión de memoria

Fases de la creación del ejecutable

Tipos de librerías

Compilación en detalle

Reubicación

Librerías de enlace estático y dinámico

¿Más preguntas?

- 1 Gestión de memoria
- 2 Fases de la creación del ejecutable
- 3 Reubicación
- 4 Librerías de enlace estático y dinámico
- 5 ¿Más preguntas?



Fases de la creación del ejecutable; compilación

CONTENIDOS

Gestión de memoria

Fases de la creación del ejecutable

Tipos de librerías

Compilación en detalle

Reubicación

Librerías de enlace estático y dinámico

¿Más preguntas?

La creación del ejecutable (a veces llamada **compilación** en general) se compone de la compilación propiamente dicha y el enlazado.

Compilación en tres fases:

- La fase de preprocesador incluye en el programa escrito en Lenguaje de alto nivel (fuentes `.c`) más código en C o modificaciones (`#include #define #ifdef`) y produce un nuevo fichero en C
- La fase de compilación produce código ensamblador
`gcc -S -masm=intel -fverbose-asm hola.c`
Para desensamblar:
`objdump -d --disassembler-options=intel hola`
- La fase de ensamblador produce código objeto (binario) con extensión `.o`



Fases de la creación del ejecutable: enlazado

CONTENIDOS

Gestión de memoria

Fases de la creación del ejecutable

Tipos de librerías

Compilación en detalle

Reubicación

Librerías de enlace estático y dinámico

¿Más preguntas?

Enlazado (montar, *linkar*, usar el *linker*) produce el ejecutable. El enlazador empareja referencias no resueltas dentro de un objeto (variables y funciones externas, librerías).

Más información y una agrupación alternativa de las fases:

<https://www.tenouk.com/ModuleW.html>



Tipos de librerías

CONTENIDOS

Gestión de memoria

Fases de la creación del ejecutable

Tipos de librerías
Compilación en detalle

Reubicación

Librerías de enlace estático y dinámico

¿Más preguntas?

Metáfora de librería (mueble que se usa para poner libros) o biblioteca (lugar al que va para consultar libros).

Tipos de librerías:

- En Lenguaje de alto nivel (fuentes .c). Se compilan a código objeto (binario con extensión .o)
- Se incluyen los ficheros objeto al enlazar.
 - Varios fuentes en un objeto
 - Varios fuentes en sendos objetos que se enlazan



Compilación en detalle (1)

CONTENIDOS

Gestión de memoria

Fases de la creación del ejecutable

Tipos de librerías

Compilación en detalle

Reubicación

Librerías de enlace estático y dinámico

¿Más preguntas?

Compilación en detalle (1)

```
1 $ gcc hola.c -o hola -v
Configured with: --enable-languages=c,ada,c++,java,go,d,fortran,objc,obj-c++ --enable-threads=posix --host=x86_64-linux-gnu --target=x86_64-linux-gnu
Thread model: posix
4 COLLECT_GCC_OPTIONS='-o' 'hola' '-v' '-mtune=generic' '-march=x86-64'
/usr/lib/gcc/x86_64-linux-gnu/5/cc1 -quiet -v -imultiarch x86_64-linux-gnu hola.c -quiet -dumpbase hola.c -mtune=generic -march=x86-64 -auxbase hola -version -fstack-protector-strong -Wformat -Wformat-security -o /tmp/ccaBngmA.s
7 GNU C11 (Ubuntu 5.2.1-22ubuntu2) version 5.2.1 20151010 (x86_64-linux-gnu)
...
```



Compilación en detalle (2)

CONTENIDOS

Gestión de memoria

Fases de la creación del ejecutable

Tipos de librerías

Compilación en detalle

Reubicación

Librerías de enlace estático y dinámico

¿Más preguntas?

```
as -v --64 -o /tmp/cczgf6hl.o /tmp/ccaBngmA.s
GNU ensamblador versión 2.25.1 (x86_64-linux-gnu)
3 LIBRARY_PATH=/usr/lib/gcc/x86_64-linux-gnu/5:/usr/lib/gcc/
x86_64-linux-gnu/5/../../../../x86_64-linux-gnu:/usr/lib/
gcc/x86_64-linux-gnu/5/../../../../lib:/lib/x86_64-
linux-gnu:/lib/./lib:/usr/lib/x86_64-linux-gnu:/usr
/lib/./lib:/usr/lib/gcc/x86_64-linux-gnu
/5/../../../../lib:/usr/lib/
/usr/lib/gcc/x86_64-linux-gnu/5/collect2 -plugin /usr/lib/
gcc/x86_64-linux-gnu/5/liblto_plugin.so -m elf_x86_64
6 --as-needed -dynamic-linker /lib64/ld-linux-x86-64.so.2 -z
relro -o hola
/usr/lib/gcc/x86_64-linux-gnu/5/../../../../x86_64-linux-gnu/
crt1.o /usr/lib/gcc/x86_64-linux-gnu/5/crtend.o /usr/
lib/gcc/x86_64-linux-gnu/5/../../../../x86_64-linux-gnu/
crtn.o
```

http://wiki.osdev.org/Calling_Global_Constructors



CONTENIDOS

Gestión de memoria

Fases de la creación del ejecutable

Reubicación

Tipos de Reubicación
Ejecutable ELF

Librerías de enlace estático y dinámico

¿Más preguntas?

- 1 Gestión de memoria
- 2 Fases de la creación del ejecutable
- 3 Reubicación**
- 4 Librerías de enlace estático y dinámico
- 5 ¿Más preguntas?



CONTENIDOS

Gestión de memoria

Fases de la creación del ejecutable

Reubicación

Tipos de Reubicación

Ejecutable ELF

Librerías de enlace estático y dinámico

¿Más preguntas?

Los programas deben ser *colocados* en memoria en diferentes direcciones para poder tener programas concurrentes.

- El compilador (re)ubica *en tiempo de compilación*. Código inamovible a no ser que el cargador busque y recalculé todas las referencias a memoria en todo el código binario.
- El compilador prepara el código para cargar en la dirección 0 y se reubica *en tiempo de carga* en memoria.
- Cuando el programa funciona como si tuviera toda la memoria para sí, y la CPU reubica/resuelve las direcciones *en tiempo de ejecución*.



De código fuente a código objeto

CONTENIDOS

Gestión de memoria	2
Fases de la creación del ejecutable	5
Reubicación	
Tipos de Reubicación Ejecutable ELF	8
Librerías de enlace estático dinámico	11
¿Más preguntas?	

```
...
int Global1, x, Resul;
main()
{
Global1 = 35;
x = 3;
Resul = Global1 + x*x;
escribir(Resul);
}
void escribir(int R)
{
printf("El resultado es %d\n", R);
}
```

1	0: GL
	4: x:
	8: Re
4	PROC m
	12: PU
	16: MO
7	20: MO
	24: ST
	28: MO
10	32: ST
	36: LD
	40: LD
13	44: MU
	48: AD
	52: ST
16	56: PU
	60: CA
	...
19	128: R
	PROC e
	132: P
22	.. 14/24



De código fuente a código objeto

CONTENIDOS

Gestión de memoria

Fases de la creación del ejecutable

Reubicación

Tipos de Reubicación
Ejecutable ELF

Librerías de enlace estático y dinámico

¿Más preguntas?

```
1 0: Global1:
4: x:
8: Resul:
4 PROC main
12: PUSH r30      ;; main
16: MOV r30,r31
7 20: MOV r1, 35
24: ST r1, @Global1
28: MOV r1, 3
10 32: ST r1, @x
36: LD r2, @Global1
40: LD r3 ,@x
13 44: MULT r4, r3,r3
48: ADD r5, r2,r4
52: ST r5, @Resul
16 56: PUSH r5
60: CALL escribir
...
19 128: RET
PROC escribir
132: PUSH r30 ;; escribir
22 ...
252: RET
```



Ejecutable ELF

CONTENIDOS

Gestión de memoria

Fases de la creación del ejecutable

Reubicación

Tipos de Reubicación
Ejecutable ELF

Librerías de enlace estático y dinámico

¿Más preguntas?

Elementos del ejecutable:

- Tabla de reubicación (lugar dentro del ejecutable de valores @Global1, @Resul y escribir)
- Tamaño del código binario
- Byte de la primera instrucción
- instrucciones binarias, Lenguaje Máquina
- Datos del programa integrados en el ejecutable

file nombrededefichero (info del fichero, *magic numbers*)

https://en.wikipedia.org/wiki/Executable_and_Linkable_Format



CONTENIDOS

Gestión de memoria

Fases de la creación del ejecutable

Reubicación

Librerías de enlace estático y dinámico

¿Más preguntas?

- 1 Gestión de memoria
- 2 Fases de la creación del ejecutable
- 3 Reubicación
- 4 Librerías de enlace estático y dinámico
- 5 ¿Más preguntas?



Librerías de enlace estático y dinámico

CONTENIDOS

Gestión de memoria

Fases de la creación del ejecutable

Reubicación

Librerías de enlace estático y dinámico

¿Más preguntas?

Las librerías de enlace estático se incluyen en el ejecutable (ocupan disco) cuando se enlazan, y cada vez que se carga un programa con una librería estática dicha librería se replica en memoria (ocupan memoria y tiempo de carga duplicado).

Librerías de Enlace Dinámico, *Dynamic Link Libraries*, (*DLL*), *Shared Objects* (*SO*):

- Sólo hay una copia en el disco, fuera de los ejecutables
- Se cargan bajo pedido al SO mediante una llamada al sistema
- El SO gestiona su carga y su borrado de RAM

Se ejecutan a través de una rutina de enlace, que pide la carga al SO.



Ventajas de las Librerías de enlace dinámico

CONTENIDOS

Gestión de
memoria

Fases de la
creación del
ejecutable

Reubicación

Librerías de
enlace estático y
dinámico

¿Más preguntas?

Ventajas de las Librerías de enlace dinámico:

- Se ocupa menos disco, los ejecutables son más pequeños
- Se ocupa menos memoria, sólo una copia por librería
- Se pueden hacer actualizaciones sin recompilar el programa principal
- Se pueden ejecutar programas más grandes que la RAM (*overlays*)
- Se puede actualizar el SO sin reiniciar ni parar



Inconvenientes de las Librerías de enlace dinámico

CONTENIDOS

Gestión de memoria

Fases de la creación del ejecutable

Reubicación

Librerías de enlace estático y dinámico

¿Más preguntas?

Inconvenientes de las Librerías de enlace dinámico:

- El mecanismo de llamada es algo más lento
- La gestión de dependencias entre DLLs puede ser complejo
- Es necesario un control de versiones



Librerías dinámicas enlazadas

CONTENIDOS

Gestión de memoria

Fases de la creación del ejecutable

Reubicación

Librerías de enlace estático y dinámico

¿Más preguntas?

```
1 $ ldd hola
linux-vdso.so.1 => (0x00007fffd1358f000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0
x00007f06b4e02000)
4 /lib64/ld-linux-x86-64.so.2 (0x000055b243e7f000)

$ ldd libstaticapi
7 no es un ejecutable dinámico

$ ldd libdynamicapi
10 linux-vdso.so.1 (0x00007ffefc7d1000)
libpi.so.1 => ./libpi.so.1 (0x00007f25c885d000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0
x00007f25c861b000)
13 /lib64/ld-linux-x86-64.so.2 (0x00007f25c8869000)
```

http://wiki.osdev.org/Calling_Global_Constructors



CONTENIDOS

Gestión de memoria

Fases de la creación del ejecutable

Reubicación

Librerías de enlace estático y dinámico

¿Más preguntas?

¿Más preguntas?

- 1 Gestión de memoria
- 2 Fases de la creación del ejecutable
- 3 Reubicación
- 4 Librerías de enlace estático y dinámico
- 5 ¿Más preguntas?



¿Más preguntas?

CONTENIDOS

Gestión de memoria

Fases de la creación del ejecutable

Reubicación

Librerías de enlace estático y dinámico

¿Más preguntas?

¿Más preguntas?

¿Más preguntas?



5 Desarrollo y Carga de
Programas de C en Memoria:
Librerías estáticas y dinámicas
Introducción a los Sistemas Operativos,
2021-2022

Pablo González Nalda

Depto. de Lenguajes y Sistemas Informáticos
EU de Ingeniería de Vitoria-Gasteiz,
UPV/EHU



8 de marzo de 2022