

LIBRERÍAS COMPARTIDAS



Librerías Compartidas

■ Resolución de Símbolos

1. Resolver símbolos locales es fácil
2. Para símbolos globales, se asume están en otro modulo

¿Y si símbolo está definido muchas veces?

Librerías Compartidas

Símbolo:

- Fuertes (funciones, variables globales inicializadas)
- Débiles (variables globales no inicializadas)

Liberías Compartidas

Reglas del Linker:

1. No se permiten multiples símbolos fuertes
2. Entre símbolo débil y uno fuerte, se elige el fuerte
3. Entre varios símbolos débiles, se elige cualquiera de ellos.

Librerías Compartidas

Linking Estático

Mecanismo para “empacar” archivos objeto re-ubicables en un solo ejecutable.

¿Mecanismos?

Librerías Compartidas

■ Solución:

Librerías estáticas.

1. Agrupa funciones relacionadas en un solo archivo
2. Al hacer linking, solo se copian funciones necesarias.

Archivo: colección de archivos objeto reubicables concatenados (.a)

Librerías Compartidas

- Para crear librería .a desde varios .c:

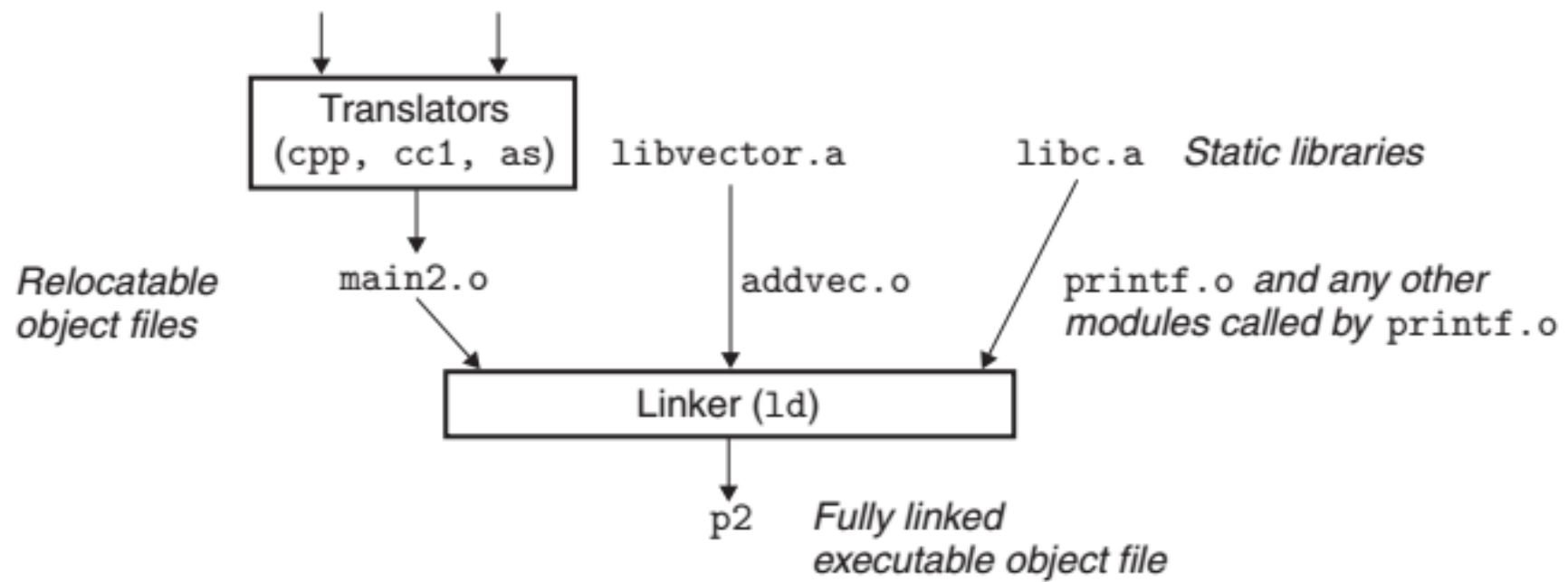
```
gcc -c *.c
```

```
ar rcs nombre.a *.o
```

Para hacer linking con dicha librería:

```
gcc *.o *.a -o nombre
```

Source files main2.c vector.h



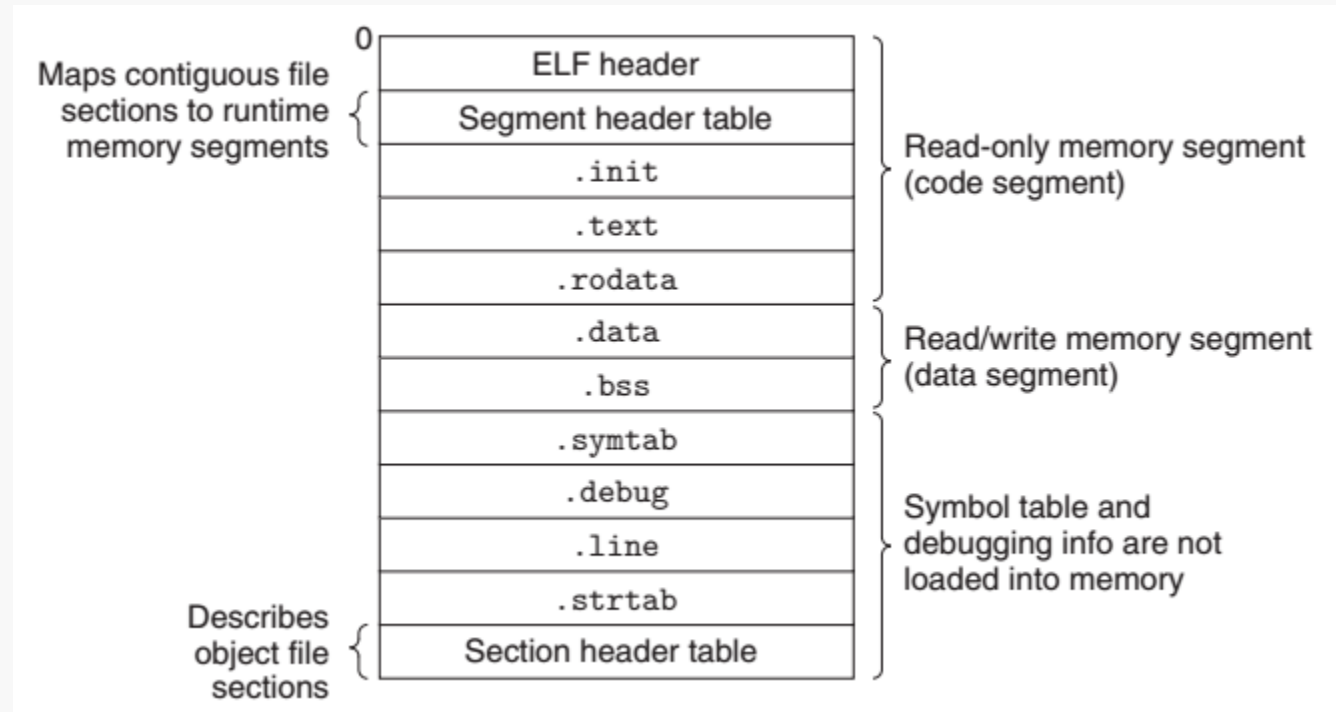
Librerías Compartidas

- Siempre el archivo re-ubicable donde se define símbolo debe estar **después** de donde se lo usa (en comando gcc).
- Se puede repetir el nombre de la librería varias veces en caso de inter-dependencias. Ej (asuma foo.c depende libx y liby de libx):

gcc foo.c libx.a liby.a libx.a

Librerías Compartidas

Ejecutando archivo objeto reubicables



Librerías Compartidas

Para ejecutarlos:

./programa

El programa se empieza a ejecutar gracias al **loader**

Loader: Copia el código y datos del archivo objeto ejecutable a la memoria, y salta al **entry point** (la primera instrucción).

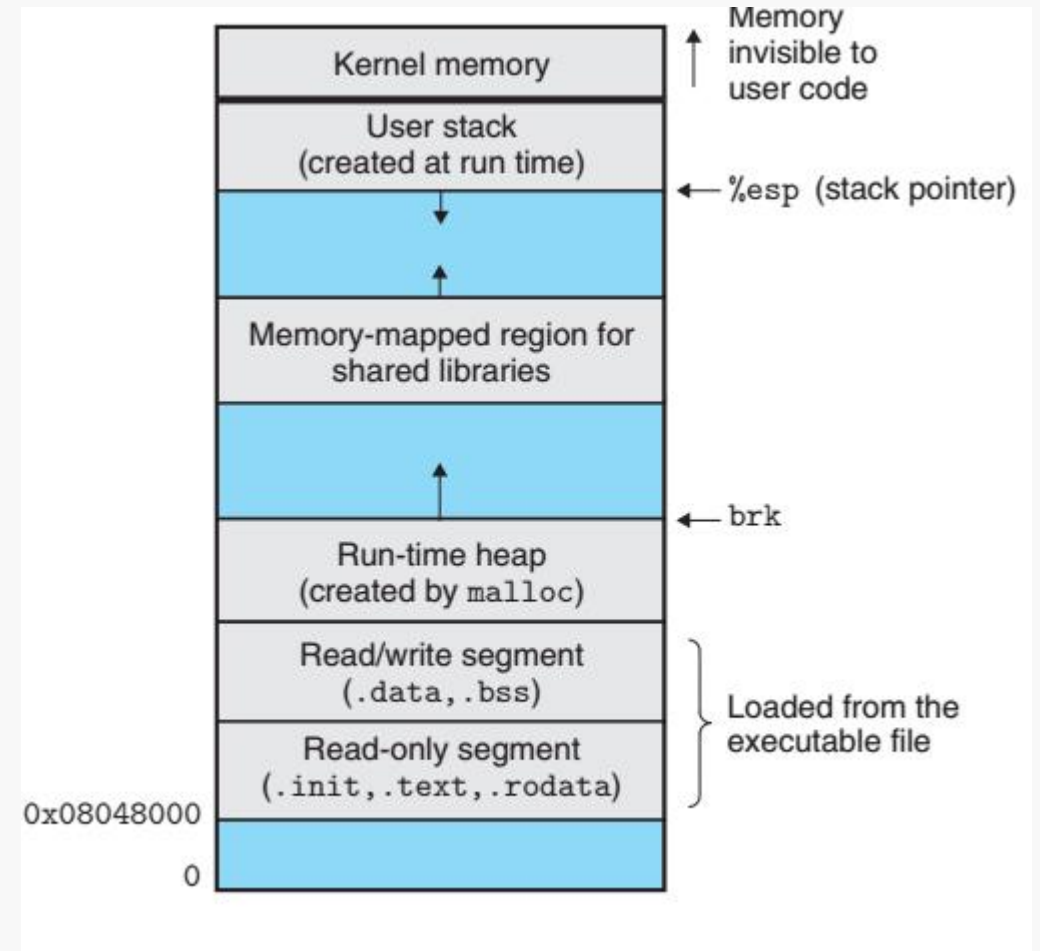
Librerías Compartidas

Objeto ejecutable en memoria

Básicamente, un proceso.

Varias secciones:

- Stack, heap
- Secciones de datos y código



Librerías Compartidas

- Cuando loader se ejecuta:
 1. Copia segmentos de datos y código a memoria
 2. Loader salta al **entry point** (dirección _start)
 3. El código de inicio es igual para todo programa de C

Librerías Compartidas

Linking dinámico

Librerías compartidas pueden ser cargadas al momento de ejecutar el programa en cualquier dirección de memoria, y enlazada con un programa *en memoria*.

Librerías Compartidas

Linking dinámico

Librerías compartidas pueden ser cargadas al momento de ejecutar el programa en cualquier dirección de memoria, y enlazada con un programa *en memoria*.

Objetos compartidos → extensión *.so*

Librerías Compartidas

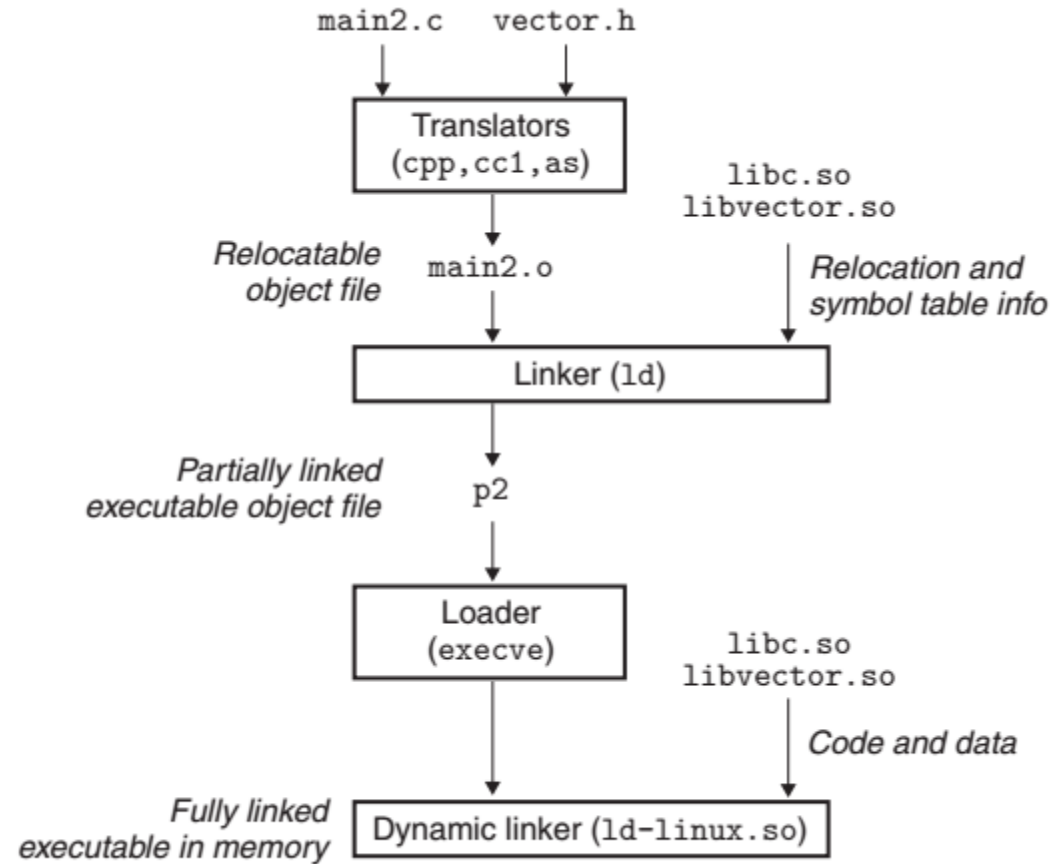
Se comparten de dos maneras:

1. Todos los programas usan el mismo archivo .so
2. Solo una instancia de la sección de texto se carga en memoria.

Para generar .so usamos banderas:

-shared y -fPIC

Librerías Compartidas



Librerías Compartidas

Cargando librerías dinámicas durante ejecución

Usamos:

```
#include <dlfcn.h>
```

```
void *dlopen(const char* filename, int flag);
```

Banderas : RTLD_ GLOBAL , RTLD_NOW, RTLD_LAZY

Librerías Compartidas

`int dlclose (void *handle);` → Desacarga la librería

`int dlclose (void *handle);` → Nos devuelve errores generados por `dlsym` o `dlopen`

`void *dlsym(void *handle, char *symbol);` → Nos devuelve la dirección del símbolo `symbol`.