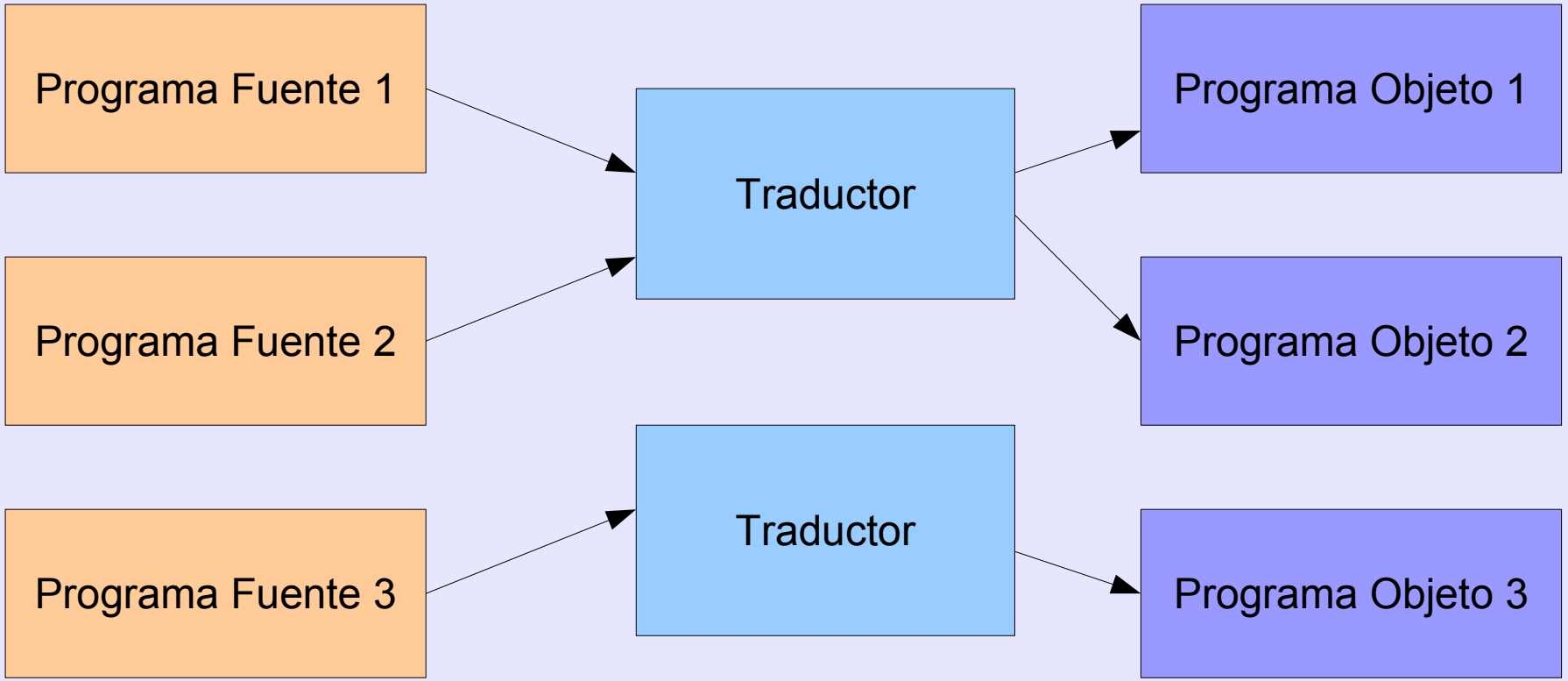
75-08 Sistemas Operativos
Lic. Ing. Osvaldo Clúa
2008

Facultad de Ingeniería
Universidad de Buenos Aires

Linkers

Uso del Linker
(a) Traducción

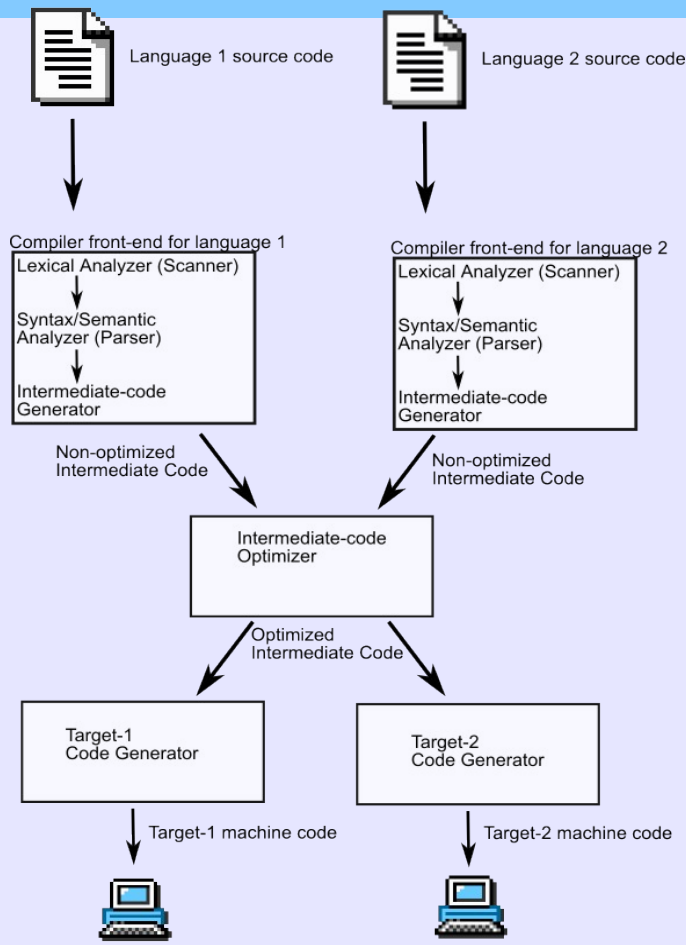


Traducción - ensamblado

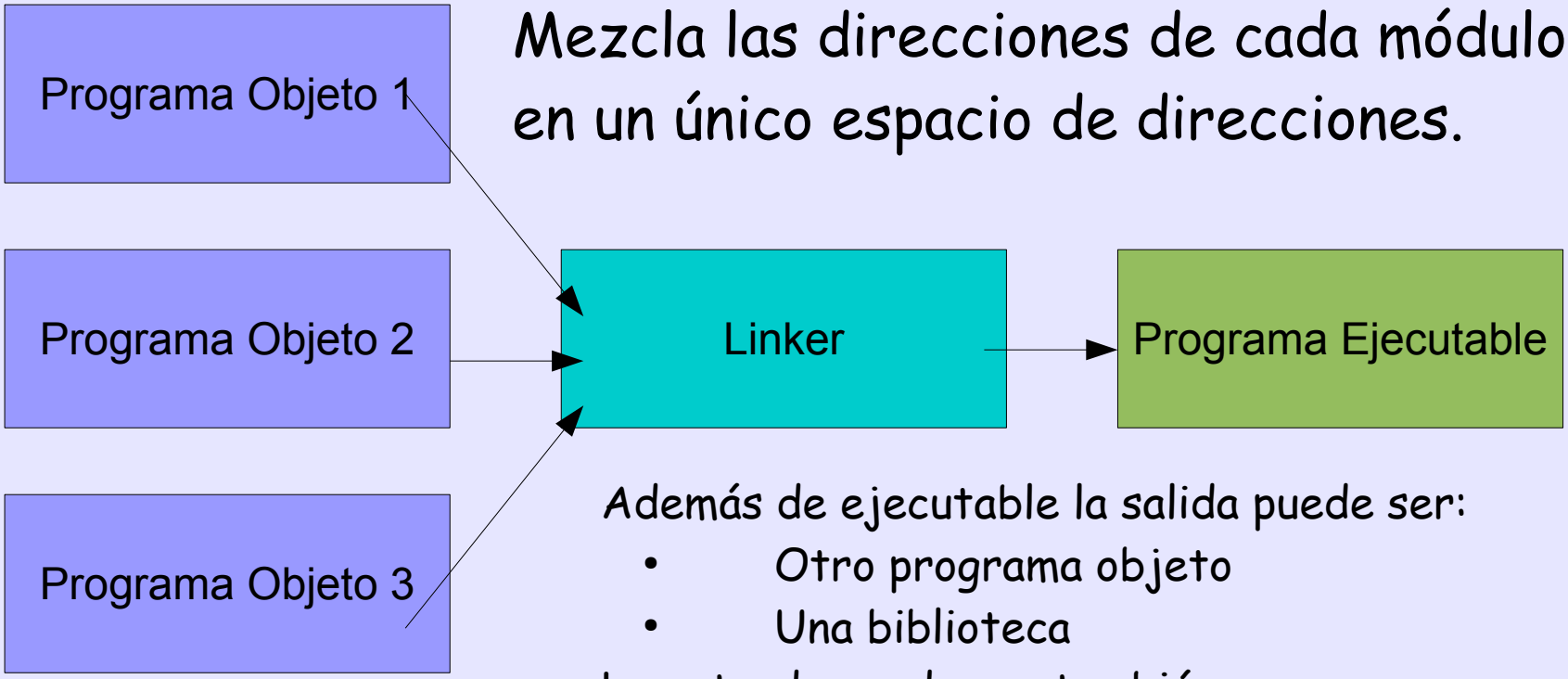
- Si el lenguaje es un assembler, la traducción es un ensamblado (assembly) hecho por un programa ensamblador (assembler).
  - Convierte código de lenguaje ensamblador memotécnico a códigos de operación.
  - Resuelve identificadores a posiciones de memoria.
  - Algunos proveen abstracciones de programación avanzadas.
  - Existe un assembler distinto para cada arquitectura, incluso hay assemblers generales

Traducción - Compilación

- Si se trata de un lenguaje de alto o mediano nivel, la traducción es una compilación.
- Un Libro online sobre compiladores del curso de Leonidas Fegaras.



Link-Editor



Mezcla las direcciones de cada módulo en un único espacio de direcciones.

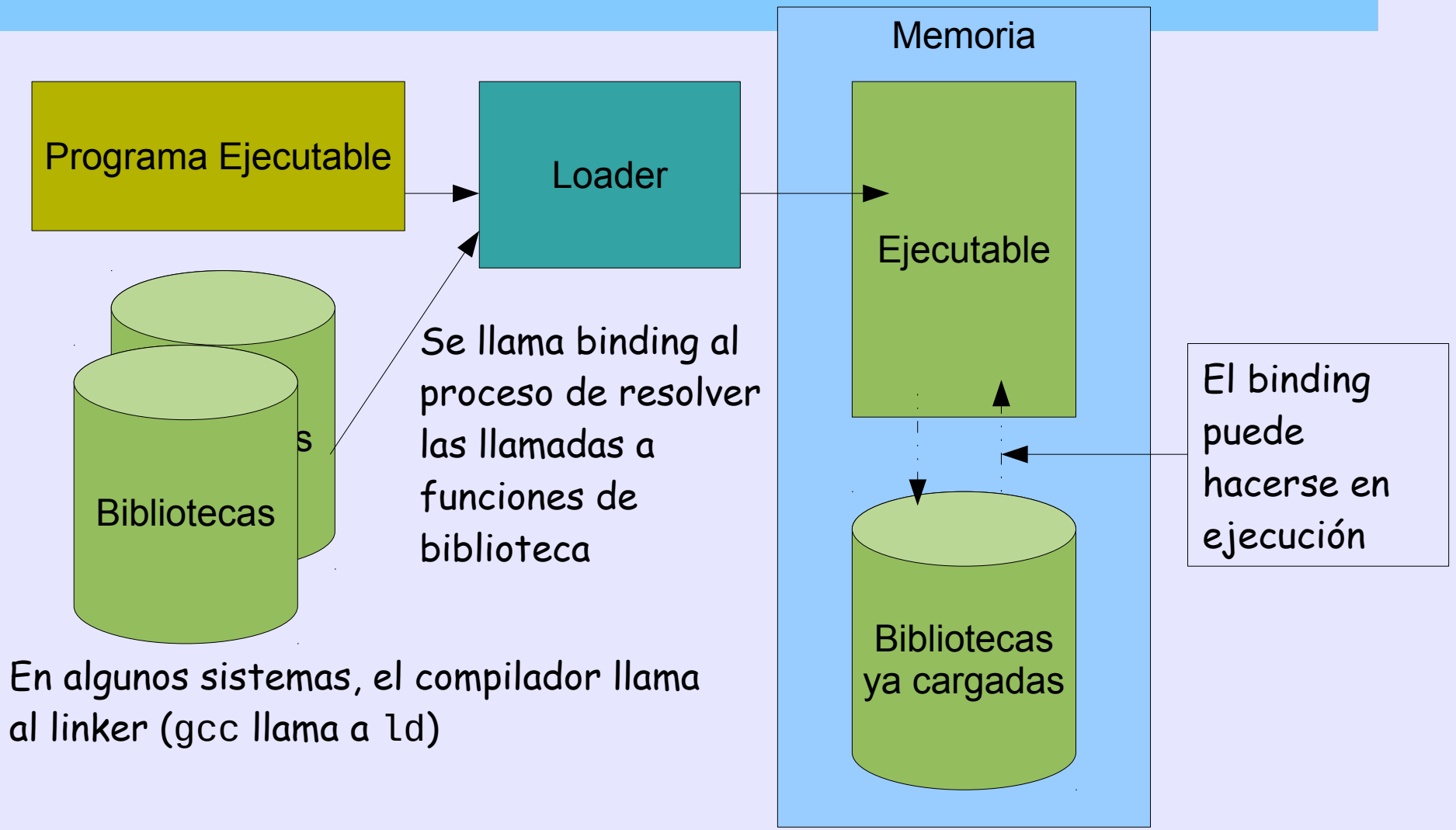
Además de ejecutable la salida puede ser:

- Otro programa objeto
- Una biblioteca

La entrada puede ser también

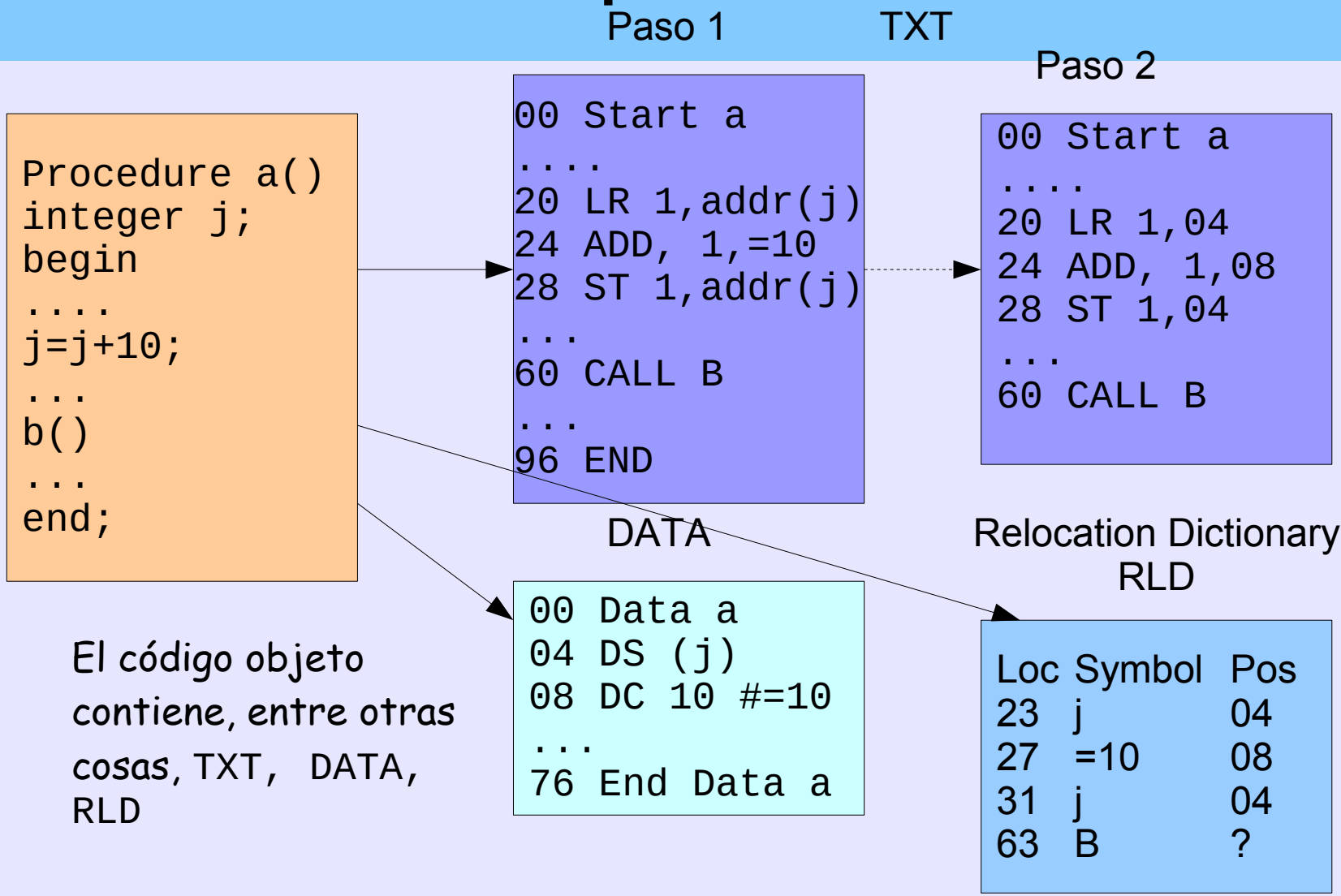
- Un ejecutable
- Una biblioteca

Loader

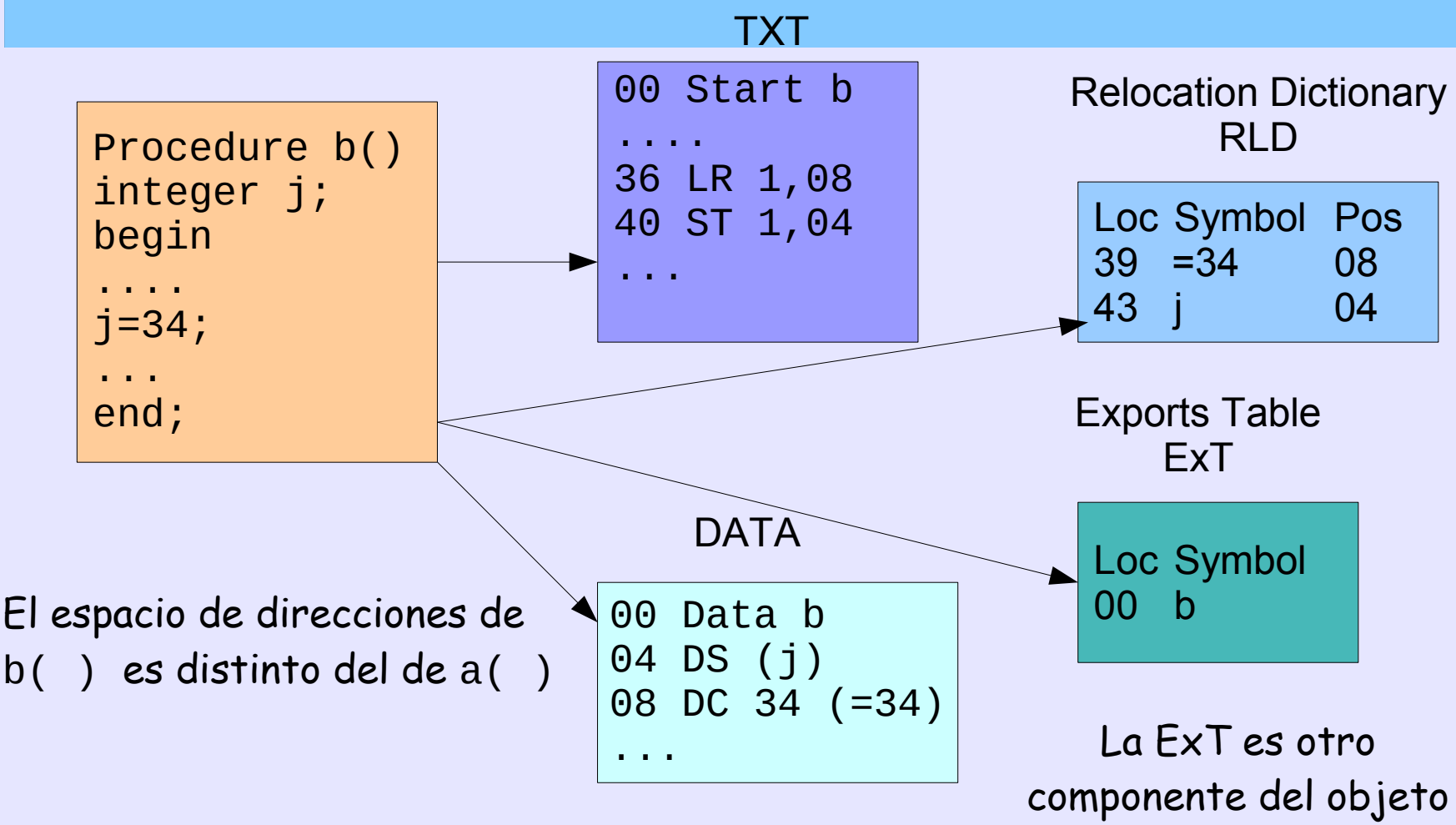


En algunos sistemas, el compilador llama al linker (gcc llama a `ld`)

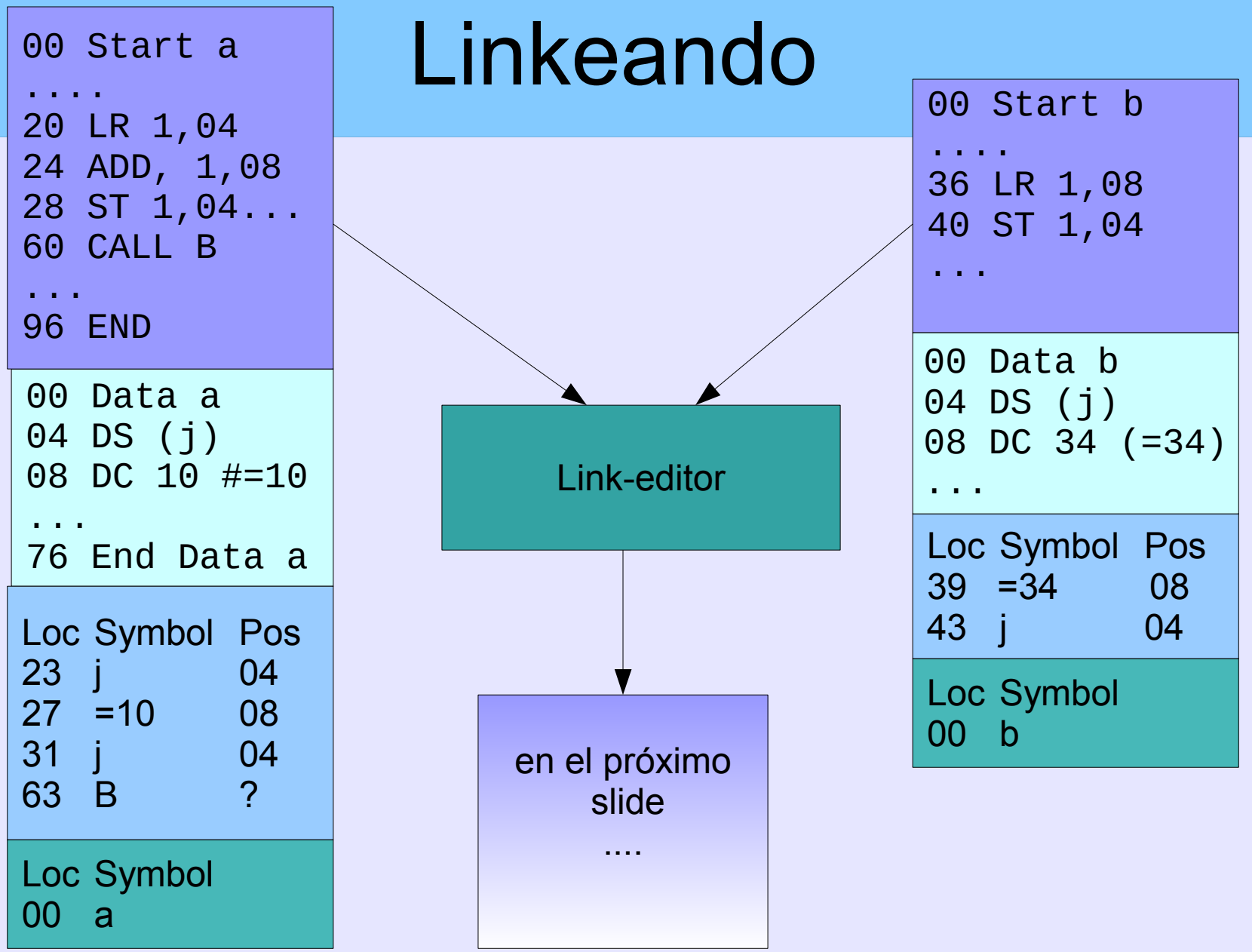
Compilando


Paso 1
TXT
Paso 2
Procedure a()
integer j;
begin
....
j=j+10;
...
b()
...
end;
00 Start a
....
20 LR 1,addr(j)
24 ADD, 1,=10
28 ST 1,addr(j)
...
60 CALL B
...
96 END
00 Start a
....
20 LR 1,04
24 ADD, 1,08
28 ST 1,04
...
60 CALL B
DATA
Relocation Dictionary
RLD
00 Data a
04 DS (j)
08 DC 10 #=10
...
76 End Data a
Loc Symbol Pos
23 j 04
27 =10 08
31 j 04
63 B ?
El código objeto contiene, entre otras cosas, TXT, DATA, RLD

Compilando (2)



Linkeando


00 Start a
....
20 LR 1,04
24 ADD, 1,08
28 ST 1,04...
60 CALL B
...
96 END
00 Data a
04 DS (j)
08 DC 10 #=10
...
76 End Data a
Loc Symbol Pos
23 j 04
27 =10 08
31 j 04
63 B ?
Loc Symbol
00 a
00 Start b
....
36 LR 1,08
40 ST 1,04
...
00 Data b
04 DS (j)
08 DC 34 (=34)
...
Loc Symbol Pos
39 =34 08
43 j 04
Loc Symbol
00 b
Link-editor
en el próximo slide
....

Linkeando (2)

TXT

```
00 Start a
....
20 LR 1,04
24 ADD, 1,08
28 ST 1,04...
60 CALL 100
...
96 END
```

```
100 Start b
....
136 LR 1,88
140 ST 1,84
...
```

Se puede resolver el llamado a B()

DATA

```
00 Data a
04 DS (j)
08 DC 10 #=10
...
76 End Data a
```

```
80 Data b
84 DS (j)
88 DC 34 (=34)
...
```

El linker mezcló los espacios de direcciones de los dos procedimientos en uno solo (relocación)

RLD

Loc	Symbol	Pos
23	j	04
27	=10	08
31	j	04
63	B	100
139	=34	88
143	j	84

ExT

Loc	Symbol
00	a
100	b

Object file formats (OFF)

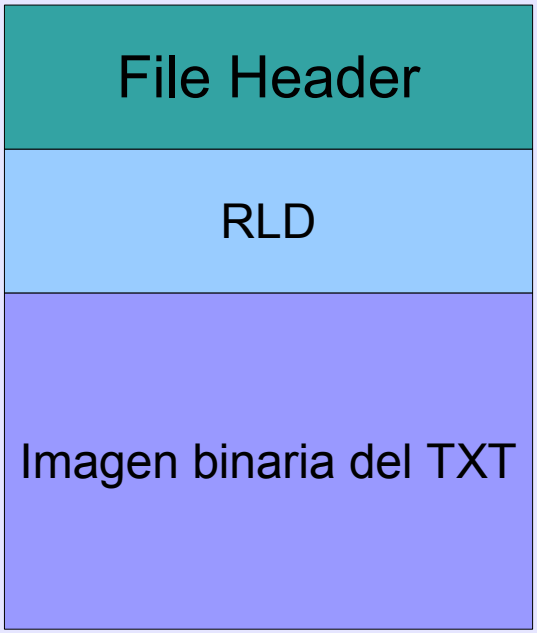
- Es clave para la performance del sistema.
- Algunos preveen su interacción con el paginado.
- Se suele utilizar el mismo formato para:
  - Ejecutables
  - Objetos
  - Bibliotecas
- Que dos Sistemas Operativos tengan el mismo OFF no significa que los programas de uno puedan correr en el otro.

DOS com

- Son los que tienen extensión .com. Microsoft los llama bin o binary file
- Se hacen con exe2bin.
- Se cargan en una dirección fija de memoria (0x100).
- Datos y código estan en el mismo segmento.
- Su tamaño máximo es de 65,280 (0xFF00) bytes
- Se puede decir que es un "null file format"

DOS exe

- Aparece en DOS 2.0.
- Su primer byte o (Magic number) es "MZ", iniciales de Mark Zbikowski.
- Tiene previsión para relocación en memoria..



Header del DOS .exe

```
char signature[2] = "MZ"; // magic number
short lastsize; // # bytes used in last block
short nblocks; // number of 512 byte blocks
short nreloc; // number of relocation entries
short hdrsize; // size of file header in 16 byte paragraphs
short minalloc; // minimum extra memory to allocate
short maxalloc; // maximum extra memory to allocate
void far *sp; // initial stack pointer
short checksum; // ones complement of file sum
void far *ip; // initial instruction pointer
short relocpos; // location of relocation fixup table
short noverlay; // Overlay number, 0 for program
char extra[]; // extra material for overlays, etc.
void far *relocs[]; // relocation entries, starts at relocpos
```

Common Object File Format
COFF

- Aparece en Unix pero se usa en otros ambientes.
- Se compone de varias secciones separadas por headers (con limitación de longitud).
- Se usa para bibliotecas.
  - Aunque no de enlace dinámico. AIX usa XCOFF
- Soporta debug (pero solo de C).
- nm(1) lo puede inspeccionar.

Windows Portable Executable (PE)

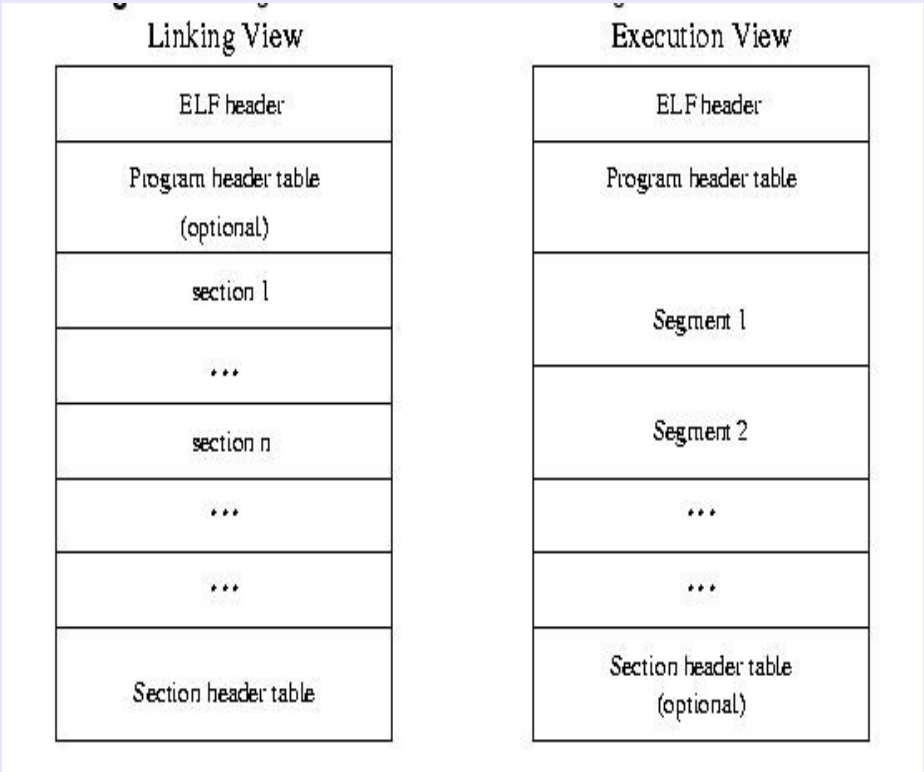
- Es una adaptación del COFF. El Windows hace un wrapping del COFF.
- Se usa en Intel, ARM y SuperH (Windows CE).
- Además Intel lo adoptó para EFI.
- Microsoft también lo usa en .net para la máquina virtual Common Language Runtime .
- Mono lo debió adoptar (quiere ser compatible a nivel binario con .net).

PE de Microsoft

- Su magic es "PE" pero comienza con un "MZ" por "compatibilidad".
- Tiene definido espacio para resources.
- Tiene definido tablas para el uso de bibliotecas compartidas.
- Hay herramientas de análisis.
- En 64 bits se lo conoce como PE+ o PE32+

Executable and Linkable Format (ELF)

- Sirve para ejecutables y bibliotecas.
- Un directorio permite agregar nuevas secciones.
- Tiene previsiones para emulación
  - Linux, Solaris, IRIX, FreeBSD, NetBSD, OpenBSD, PlayStation



Referencias: La especificación y man elf(5).

Binarios Universales

- Mac usa la idea en 2005 para facilitar el pase de PPC a Mactel.
  - Ya la había usado en al pasar de 68k a PPC.
- Se basa en el concepto de fat binary. El formato es Mach-O.
  - Puede usar un emulador: Rosetta
  - Puede tener código ppc-32,ppc-64 y x86-64 EM64T