

Fundamentos de los Sistemas Operativos (FSO)

Departamento de Informàtica de Sistemes y Computadoras (DISCA)
Universitat Politècnica de València

Bloque Temático 4: Gestión de Memoria
Seminario 9

Mapa de memoria de un proceso
Linux

fSO

DISCA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

- **Objetivos**

- Comprender el **concepto de mapa de memoria** de un proceso
- Conocer las características del mapa de memoria en proceso Linux
- Describir la técnica para **mapear archivos en memoria**
- Diferenciar las ventajas y desventajas de trabajar con **librerías estáticas y dinámicas**

- **Bibliografía**

- “Sistemas operativos: una visión aplicada”, Capítulo 5, Jesús Carretero 2º Ed, Mac GrawHill

- **Introducción**
- Mapa de memoria de un proceso en Linux
- Archivos mapeados en memoria
- Bibliotecas de enlace dinámico

- **Mapa de memoria de un proceso**

- El S. O. gestiona el mapa de memoria de un proceso durante la vida del mismo
- El mapa de memoria es un **atributo** del proceso
- Contiene información sobre las distintas áreas del proceso:
 - Código, datos, pila ...

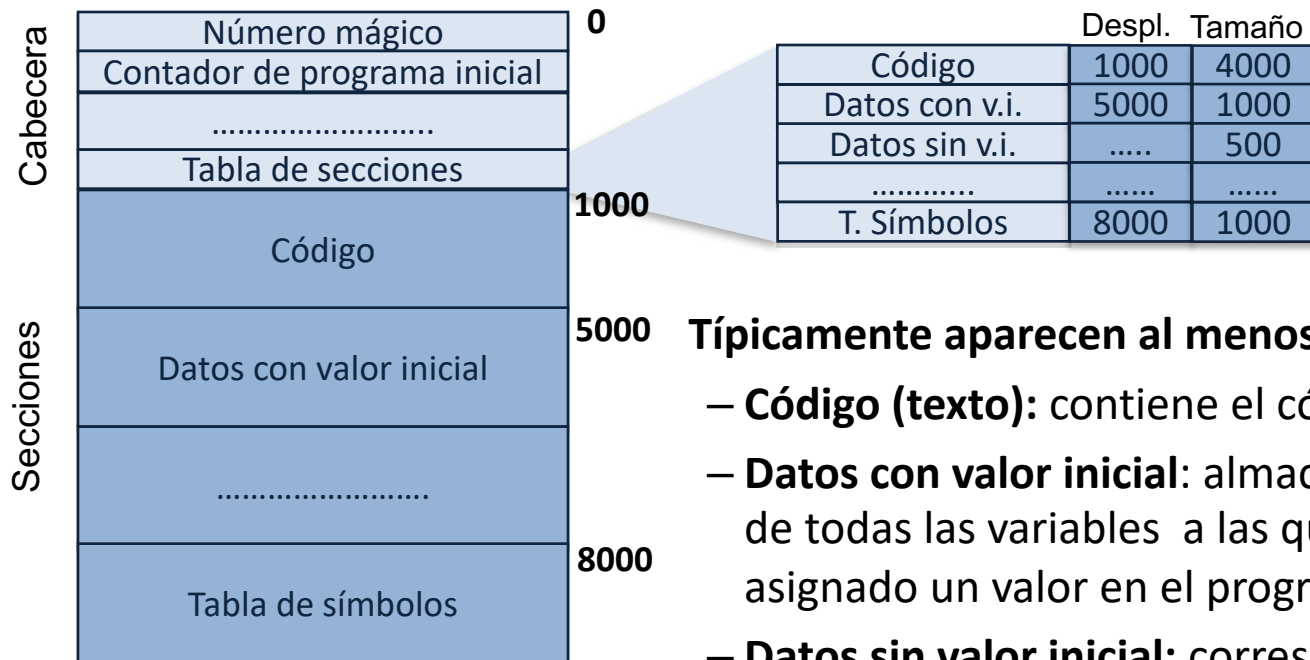
PCB



- **Mapa de memoria inicial** de un proceso: está muy vinculado con el **archivo** que contiene el código **ejecutable** de dicho proceso
- Actualmente los SO ofrecen un **modelo de memoria dinámico** mediante regiones que dan soporte a dichas áreas, de manera que durante la ejecución del proceso
 - Se pueden **crear nuevas regiones** que no obtienen los datos del fichero ejecutable: pila, archivos proyectados, memoria dinámica
 - Se pueden **eliminar regiones**

• Formato simplificado de un archivo ejecutable

- El resultado final de compilar y enlazar un programa es generar un archivo ejecutable que contenga el código máquina del programa
- Un ejecutable esta estructurado **en cabecera y un conjunto de secciones**
 - **Cabecera:** contiene información de control que permite interpretar el resto del contenido del fichero ejecutable
 - **Secciones:** Cada ejecutable tiene un conjunto de secciones



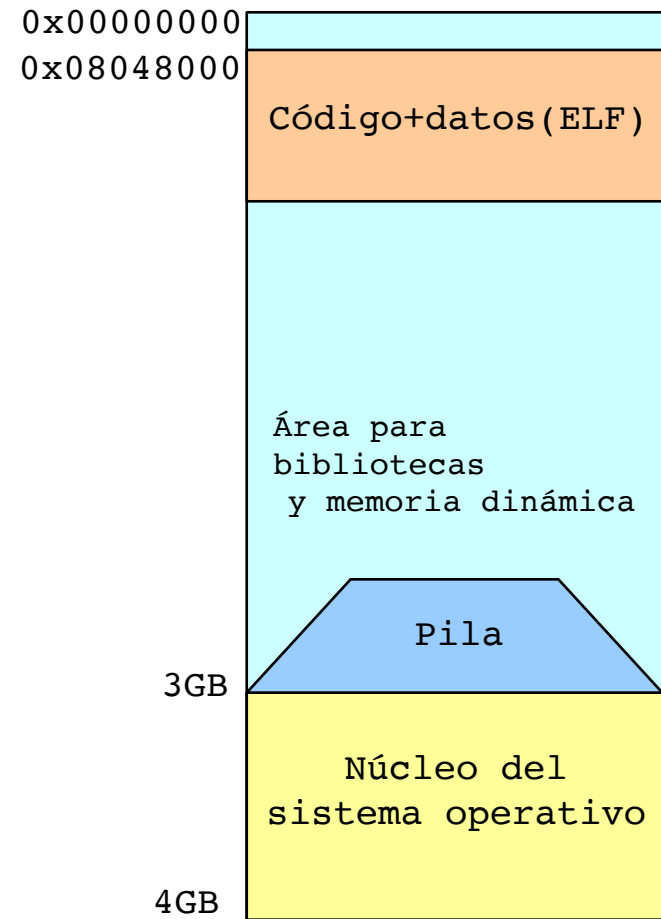
Típicamente aparecen al menos tres secciones:

- **Código (texto):** contiene el código del programa
- **Datos con valor inicial:** almacena el valor inicial de todas las variables a las que se les ha asignado un valor en el programa
- **Datos sin valor inicial:** corresponde a variables a las que no se les ha asignado un valor inicial

- Introducción
- **Mapa de memoria de un proceso Linux**
- Archivos mapeados en memoria
- Bibliotecas de enlace dinámico

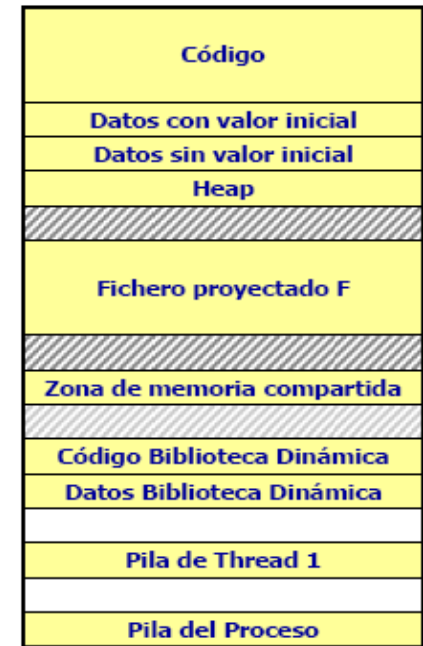
- Mapa de memoria Linux en **versiones de Linux 2.6.x** sobre arquitectura de 32 bits:

- Espacio lógico de 4GB (2^{32})
- 1er Gbyte :Código + datos inicializados que empiezan en la dirección 0x08048000 (formato ELF)
- 3er Gbyte para pila: la pila crece hacia posiciones decreciente y empieza en la parte alta del 3er GB
- 4º Gbyte: son direcciones más altas se reserva para el S.O.
- El resto del espacio puede utilizarse para datos y bibliotecas de enlace dinámico

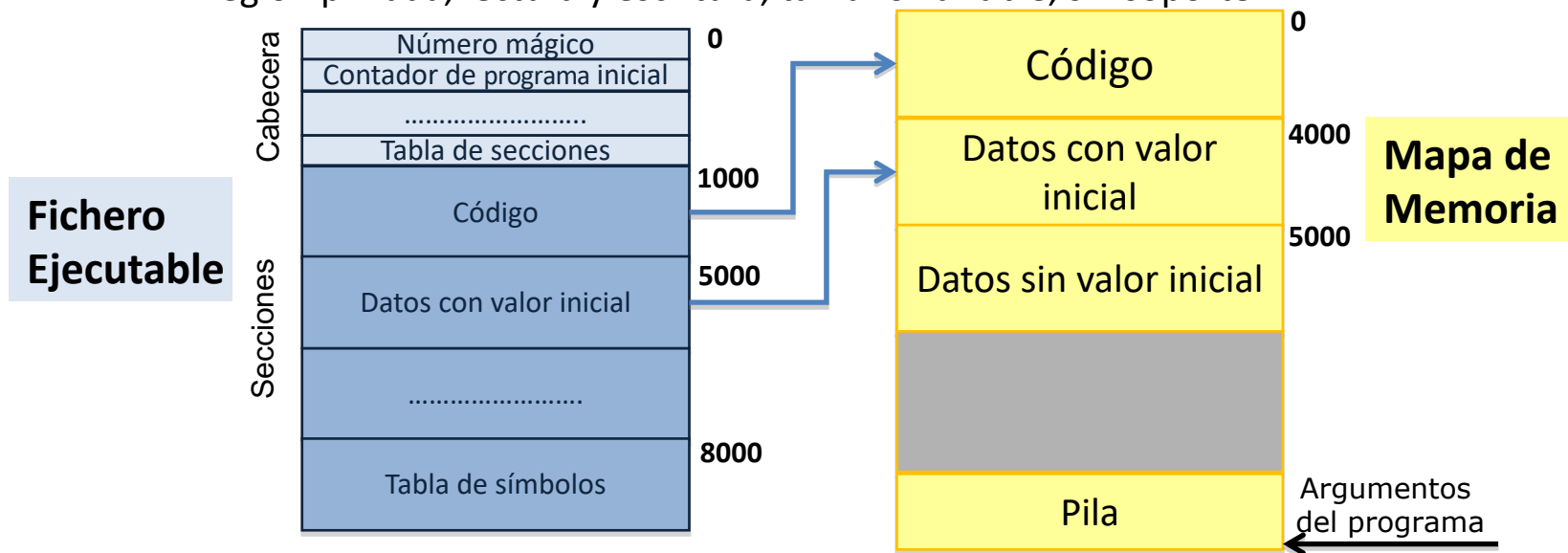


- **Mapa de memoria esta formado por regiones**

- Una región tiene asociada una determinada información
 - código, datos con valor inicial, datos sin valor inicial, ficheros proyectados, códigos bibliotecas dinámicas
- Cada región es una **zona contigua** caracterizada por la dirección dentro del mapa donde comienza y su tamaño
- Características de las regiones
 - **Soporte:** dónde se almacena la información de la región
 - **en archivo:** información de la región almacenada en fichero
 - **sin soporte:** la región no tiene contenido inicial
 - Tipo de **compartición:**
 - **Privada (p)** : contenido de la región sólo es accesible por el proceso
 - **Compartida:** contenido de la región puede ser accesible por varios procesos
 - **Protección:** Tipo de acceso permitido a esa región
 - lectura, escritura y/o ejecución
 - **Tamaño:** fijo o variable



- Al ejecutar un programa se crea un **mapa de memoria inicial** a partir del archivo ejecutable
- **Cada sección del ejecutable** da lugar a **una región del mapa inicial**
 - **Código (texto):**
 - región compartida, lectura y ejecución, tamaño fijo, soporte en archivo ejecutable
 - **Datos con valor inicial:** cada proceso necesita una copia propia de las variables.
 - región privada (), lectura y escritura, tamaño fijo, soporte en archivo ejecutable
 - **Datos sin valor inicial:** en algunos lenguajes se rellena a cero
 - región privada, lectura y escritura, tamaño fijo, sin soporte
 - **Pila:** Crece hacia direcciones bajas, inicialmente sólo contiene los argumentos de llamada al programa
 - región privada, lectura y escritura, tamaño variable, sin soporte.



- El **mapa de memoria** de un proceso **es dinámico**. Durante la ejecución del proceso se pueden crear regiones como:
 - **Región de Heap**
 - Soporte de memoria dinámica (`malloc` en C)
 - Privada, lectura y escritura, tamaño variable, sin soporte (inicializada a cero)
 - Crece hacia direcciones más altas
 - **Archivo proyectado**
 - Cuando se mapea un fichero se crea una región asociada al archivo proyectado
 - Tamaño variable, soporte en archivo
 - Protección y carácter compartido o privado especificado en la proyección
 - **Memoria compartida**
 - Región asociada a la zona de memoria compartida (para la comunicación entre procesos)
 - Compartida, tamaño variable, sin soporte (inicializada a 0)
 - Protección especificada por el programa
 - **Pilas de hilos (threads)**
 - Cada pila de hilo corresponde con una región.
 - Estas regiones constan de las mismas características que las asociadas a la pila del proceso

- Disponemos de dos opciones para **visualizar el mapa de memoria** de un proceso **en LINUX**:

- Listar el **archivo maps** del proceso ubicado en /proc/PID_del_proceso
- **\$cat /proc/PID/maps**
 - » Rango de direcciones lógicas
 - » Permisos
 - » Desplazamiento desde inicio fichero ejecutable
 - » Dispositivo
 - » Nodo-i
 - » Nombre del fichero proyectado

- Ejecutar la **orden del shell**
\$pmap PID
 - » dirección lógica base
 - » Tamaño
 - » Permisos
 - » nombre fichero proyectado

Nota: Sustituyendo PID por la variable **\$\$** se hace referencia al **proceso** que esta **en ejecución**. Ejemplo:

```
$ pmap $$
```

```
$cat /proc/$$/maps
```

Mapa de memoria de un proceso Linux

Ejemplo: orden \$pmap en arquitectura de 32 bits

pblanes@pblanes-desktop:~\$ pmap \$\$

1608: /bin/bash

00140000 32K r-x-- /lib/tls/i686/cmov/libnss_nis-2.11.1.so

00148000 4K r---- /lib/tls/i686/cmov/libnss_nis-2.11.1.so

00149000 4K rw--- /lib/tls/i686/cmov/libnss_nis-2.11.1.so

00266000 8K r-x-- /lib/tls/i686/cmov/libdl-2.11.1.so

00268000 4K r---- /lib/tls/i686/cmov/libdl-2.11.1.so

00269000 4K rw--- /lib/tls/i686/cmov/libdl-2.11.1.so

00319000 1356K r-x-- /lib/tls/i686/cmov/libc-2.11.1.so

0046c000 4K ----- /lib/tls/i686/cmov/libc-2.11.1.so

0046d000 8K r---- /lib/tls/i686/cmov/libc-2.11.1.so

0046f000 4K rw--- /lib/tls/i686/cmov/libc-2.11.1.so

00470000 12K rw--- [anon]

005b0000 4K r-x-- [anon]

007f7000 208K r-x-- /lib/libncurses.so.5.7

0082b000 4K ----- /lib/libncurses.so.5.7

0082c000 8K r---- /lib/libncurses.so.5.7

0082e000 4K rw--- /lib/libncurses.so.5.7

00cbc000 108K r-x-- /lib/ld-2.11.1.so

00cd7000 4K r---- /lib/ld-2.11.1.so

00cd8000 4K rw--- /lib/ld-2.11.1.so

00cf7000 24K r-x-- /lib/tls/i686/cmov/libnss_compat-2.11.1.so

00cfd000 4K r---- /lib/tls/i686/cmov/libnss_compat-2.11.1.so

00cfe000 4K rw--- /lib/tls/i686/cmov/libnss_compat-2.11.1.so

00f56000 40K r-x-- /lib/tls/i686/cmov/libnss_files-2.11.1.so

00f60000 4K r---- /lib/tls/i686/cmov/libnss_files-2.11.1.so

00f61000 4K rw--- /lib/tls/i686/cmov/libnss_files-2.11.1.so

00fc1000 76K r-x-- /lib/tls/i686/cmov/libnsl-2.11.1.so

00fd4000 4K r---- /lib/tls/i686/cmov/libnsl-2.11.1.so

00fd5000 4K rw--- /lib/tls/i686/cmov/libnsl-2.11.1.so

00fd6000 8K rw--- [anon]

08048000 780K r-x-- /bin/bash

0810b000 4K r---- /bin/bash

0810c000 20K rw--- /bin/bash

08111000 20K rw--- [anon]

0876e000 1368K rw--- [anon]

b75ef000 156K r---- /usr/share/locale-langpack/es/LC_MESSAGES/bash.mo

b7616000 252K r---- /usr/lib/locale/es_ES.utf8/LC_CTYPE

b7655000 4K r---- /usr/lib/locale/es_ES.utf8/LC_NUMERIC

b7656000 4K r---- /usr/lib/locale/es_ES.utf8/LC_TIME

b7657000 1144K r---- /usr/lib/locale/es_ES.utf8/LC_COLLATE

b7775000 8K rw--- [anon]

b7777000 4K r---- /usr/lib/locale/es_ES.utf8/LC_MONETARY

b7778000 4K r---- /usr/lib/locale/es_ES.utf8/LC_MESSAGES/SYS_LC_MESSAGES

b7779000 4K r---- /usr/lib/locale/es_ES.utf8/LC_PAPER

b777a000 4K r---- /usr/lib/locale/es_ES.utf8/LC_NAME

b777b000 4K r---- /usr/lib/locale/es_ES.utf8/LC_ADDRESS

b777c000 4K r---- /usr/lib/locale/es_ES.utf8/LC_TELEPHONE

b777d000 4K r---- /usr/lib/locale/es_ES.utf8/LC_MEASUREMENT

b777e000 28K r--s- /usr/lib/gconv/gconv-modules.cache

b7785000 4K r---- /usr/lib/locale/es_ES.utf8/LC_IDENTIFICATION

b7786000 8K rw--- [anon]

bf95a000 84K rw--- [stack]

total 5868K

Ejemplo: orden \$pmap en arquitectura de 64bits

pblanes@shell-sisop:~\$ pmap -d \$\$

29916: -bash

| Address | Kbytes | Mode | Offset | Device | Mapping |
|------------------|--------|-------|-------------------|-----------|----------------------|
| 0000000000400000 | 760 | r-x-- | 0000000000000000 | 008:00002 | bash |
| 00000000006bd000 | 40 | rw--- | 00000000000bd000 | 008:00002 | bash |
| 00000000006c7000 | 2616 | rw--- | 00000000006c7000 | 000:00000 | [anon] |
| 00007fa6e6728000 | 40 | r-x-- | 0000000000000000 | 008:00002 | libnss_files-2.7.so |
| 00007fa6e6732000 | 2048 | ---- | 000000000000a000 | 008:00002 | libnss_files-2.7.so |
| 00007fa6e6932000 | 8 | rw--- | 000000000000a000 | 008:00002 | libnss_files-2.7.so |
| 00007fa6e6934000 | 40 | r-x-- | 0000000000000000 | 008:00002 | libnss_nis-2.7.so |
| 00007fa6e693e000 | 2044 | ---- | 000000000000a000 | 008:00002 | libnss_nis-2.7.so |
| 00007fa6e6b3d000 | 8 | rw--- | 0000000000009000 | 008:00002 | libnss_nis-2.7.so |
| 00007fa6e6b3f000 | 88 | r-x-- | 0000000000000000 | 008:00002 | libnsl-2.7.so |
| 00007fa6e6b55000 | 2044 | ---- | 0000000000016000 | 008:00002 | libnsl-2.7.so |
| 00007fa6e6d54000 | 8 | rw--- | 0000000000015000 | 008:00002 | libnsl-2.7.so |
| 00007fa6e6d56000 | 8 | rw--- | 00007fa6e6d56000 | 000:00000 | [anon] |
| 00007fa6e6d58000 | 32 | r-x-- | 0000000000000000 | 008:00002 | libnss_compat-2.7.so |
| 00007fa6e6d60000 | 2044 | ---- | 0000000000008000 | 008:00002 | libnss_compat-2.7.so |
| 00007fa6e6f5f000 | 8 | rw--- | 0000000000007000 | 008:00002 | libnss_compat-2.7.so |
| 00007fa6e6f61000 | 1376 | r-x-- | 0000000000000000 | 008:00002 | libc-2.7.so |
| 00007fa6e70b9000 | 2048 | ---- | 00000000000158000 | 008:00002 | libc-2.7.so |
| 00007fa6e72b9000 | 12 | r---- | 00000000000158000 | 008:00002 | libc-2.7.so |
| 00007fa6e72bc000 | 8 | rw--- | 0000000000015b000 | 008:00002 | libc-2.7.so |
| 00007fa6e72be000 | 20 | rw--- | 00007fa6e72be000 | 000:00000 | [anon] |
| 00007fa6e72c3000 | 8 | r-x-- | 0000000000000000 | 008:00002 | libdl-2.7.so |
| 00007fa6e72c5000 | 2048 | ---- | 0000000000002000 | 008:00002 | libdl-2.7.so |
| 00007fa6e74c5000 | 8 | rw--- | 0000000000002000 | 008:00002 | libdl-2.7.so |

| | | | | | |
|------------------|------|-------|-------------------|-----------|---------------------|
| 00007fa6e74c7000 | 220 | r-x-- | 0000000000000000 | 008:00002 | libncurses.so.5.6 |
| 00007fa6e74fe000 | 2044 | ---- | 00000000000037000 | 008:00002 | libncurses.so.5.6 |
| 00007fa6e76fd000 | 20 | rw--- | 00000000000036000 | 008:00002 | libncurses.so.5.6 |
| 00007fa6e7702000 | 116 | r-x-- | 0000000000000000 | 008:00002 | ld-2.7.so |
| 00007fa6e77da000 | 60 | r---- | 0000000000000000 | 008:00002 | bash.mo |
| 00007fa6e77e9000 | 252 | r---- | 0000000000000000 | 008:00002 | LC_CTYPE |
| 00007fa6e7828000 | 900 | r---- | 0000000000000000 | 008:00002 | LC_COLLATE |
| 00007fa6e7909000 | 8 | rw--- | 00007fa6e7909000 | 000:00000 | [anon] |
| 00007fa6e790b000 | 4 | r---- | 0000000000000000 | 008:00002 | LC_NUMERIC |
| 00007fa6e790c000 | 4 | r---- | 0000000000000000 | 008:00002 | LC_TIME |
| 00007fa6e790d000 | 4 | r---- | 0000000000000000 | 008:00002 | LC_MONETARY |
| 00007fa6e790e000 | 4 | r---- | 0000000000000000 | 008:00002 | SYS_LC_MESSAGES |
| 00007fa6e790f000 | 4 | r---- | 0000000000000000 | 008:00002 | LC_PAPER |
| 00007fa6e7910000 | 4 | r---- | 0000000000000000 | 008:00002 | LC_NAME |
| 00007fa6e7911000 | 4 | r---- | 0000000000000000 | 008:00002 | LC_ADDRESS |
| 00007fa6e7912000 | 4 | r---- | 0000000000000000 | 008:00002 | LC_TELEPHONE |
| 00007fa6e7913000 | 4 | r---- | 0000000000000000 | 008:00002 | LC_MEASUREMENT |
| 00007fa6e7914000 | 28 | r--s- | 0000000000000000 | 008:00002 | gconv-modules.cache |
| 00007fa6e791b000 | 4 | r---- | 0000000000000000 | 008:00002 | LC_IDENTIFICATION |
| 00007fa6e791c000 | 12 | rw--- | 00007fa6e791c000 | 000:00000 | [anon] |
| 00007fa6e791f000 | 8 | rw--- | 0000000000001d000 | 008:00002 | ld-2.7.so |
| 00007fff8bc0f000 | 84 | rw--- | 00007fffffe9000 | 000:00000 | [stack] |
| 00007fff8bd1d000 | 8 | r-x-- | 00007fff8bd1d000 | 000:00000 | [anon] |
| fffffffff600000 | 4 | r-x-- | 0000000000000000 | 000:00000 | [anon] |

mapped: 21168K writeable/private: 2864K shared: 28K
gandreu@shell-sisop:~\$

- **Archivo /proc/PID/maps**

- Contiene las regiones de memoria actualmente asociadas al proceso con identificador PID y sus permisos de acceso
- El formato del archivo maps es:

| Dirección | permisos | desplaz. | disp | nodo-i | ruta |
|-----------------------|----------|----------|-------|--------|--------------------|
| 08048000-08056000 | r-xp | 00000000 | 03:0c | 64593 | /usr/sbin/gpm |
| 08056000-08058000 | rw-p | 0000d000 | 03:0c | 64593 | /usr/sbin/gpm |
| 08058000-0805b000 | rwxp | 00000000 | 00:00 | 0 | |
| 40000000-40013000 | r-xp | 00000000 | 03:0c | 4165 | /lib/ld-2.2.4.so |
| 40013000-40015000 | rw-p | 00012000 | 03:0c | 4165 | /lib/ld-2.2.4.so |
| 4001f000-40135000 | r-xp | 00000000 | 03:0c | 45494 | /lib/libc-2.2.4.so |
| 40135000-4013e000 | rw-p | 00115000 | 03:0c | 45494 | /lib/libc-2.2.4.so |
| 4013e000-40142000 | rw-p | 00000000 | 00:00 | 0 | |
| bf f f f 000-c0000000 | rwxp | 00000000 | 00:00 | 0 | |

Dirección: Espacio de direcciones que ocupa el proceso

Permisos:
r = leer
w = escribir
x = ejecutar
s = compartido
p = privado (copia en escritura)

Nodo-i: Nodo-i del dispositivo, 0 indica que no hay nodo-i asociado

Dispositivo: Es el dispositivo (mayor número : menor número)

Desplazamiento: Es el desplazamiento dentro del archivo que lo soporta

Ejemplo: archivo maps en arquitectura de 32 bits

pblanes\$ cat /proc/\$\$/maps

```
00140000-00148000 r-xp 00000000 08:01 266235 /lib/tls/i686/cmov/libnss_nis-2.11.1.so
00148000-00149000 r--p 00007000 08:01 266235 /lib/tls/i686/cmov/libnss_nis-2.11.1.so
00149000-0014a000 rw-p 00008000 08:01 266235 /lib/tls/i686/cmov/libnss_nis-2.11.1.so
00266000-00268000 r-xp 00000000 08:01 266220 /lib/tls/i686/cmov/libdl-2.11.1.so
00268000-00269000 r--p 00001000 08:01 266220 /lib/tls/i686/cmov/libdl-2.11.1.so
00269000-0026a000 rw-p 00002000 08:01 266220 /lib/tls/i686/cmov/libdl-2.11.1.so
00319000-0046c000 r-xp 00000000 08:01 266214 /lib/tls/i686/cmov/libc-2.11.1.so
0046c000-0046d000 ---p 00153000 08:01 266214 /lib/tls/i686/cmov/libc-2.11.1.so
0046d000-0046f000 r--p 00153000 08:01 266214 /lib/tls/i686/cmov/libc-2.11.1.so
0046f000-00470000 rw-p 00155000 08:01 266214 /lib/tls/i686/cmov/libc-2.11.1.so
00470000-00473000 rw-p 00000000 00:00 0
005b0000-005b1000 r-xp 00000000 00:00 0
007f7000-0082b000 r-xp 00000000 08:01 261740 /lib/libncurses.so.5.7
0082b000-0082c000 ---p 00034000 08:01 261740 /lib/libncurses.so.5.7
0082c000-0082e000 r--p 00034000 08:01 261740
0082e000-0082f000 rw-p 00036000 08:01 261740
00cbc000-00cd7000 r-xp 00000000 08:01 261663 /lib/ld-2.11.1.so
00cd7000-00cd8000 r--p 0001a000 08:01 261663 /lib/ld-2.11.1.so
00cd8000-00cd9000 rw-p 0001b000 08:01 261663 /lib/ld-2.11.1.so
```

[vdso]

/lib/libncurses.so.5.7

/lib/libncurses.so.5.7

/lib/libncurses.so.5.7

/lib/libncurses.so.5.7

/lib/ld-2.11.1.so

/lib/ld-2.11.1.so

/lib/ld-2.11.1.so

Ejemplo: archivo maps en arquitectura de 64 bits

gandreu\$ cat /proc/\$\$/maps

```
00400000-004be000 r-xp 00000000 08:02 65607
006bd000-006c7000 rw-p 000bd000 08:02 65607
006c7000-00955000 rw-p 006c7000 00:00 0
7fa6e728000-7fa6e732000 r-xp 00000000 08:02 81942
7fa6e732000-7fa6e6932000 ---p 0000a000 08:02 81942
7fa6e6932000-7fa6e6934000 rw-p 0000a000 08:02 81942
7fa6e6934000-7fa6e693e000 r-xp 00000000 08:02 81944
7fa6e693e000-7fa6e6b3d000 ---p 0000a000 08:02 81944
7fa6e6b3d000-7fa6e6b3f000 rw-p 00009000 08:02 81944
7fa6e6b3f000-7fa6e6b55000 r-xp 00000000 08:02 81939
7fa6e6b55000-7fa6e6d54000 ---p 00016000 08:02 81939
7fa6e6d54000-7fa6e6d56000 rw-p 00015000 08:02 81939
7fa6e6d56000-7fa6e6d58000 rw-p 7fa6e6d56000 00:00 0
7fa6e6d58000-7fa6e6d60000 r-xp 00000000 08:02 81940
7fa6e6d60000-7fa6e6f5f000 ---p 00008000 08:02 81940
7fa6e6f5f000-7fa6e6f61000 rw-p 00007000 08:02 81940
7fa6e6f61000-7fa6e70b9000 r-xp 00000000 08:02 81930
7fa6e70b9000-7fa6e72b9000 ---p 00158000 08:02 81930
7fa6e72b9000-7fa6e72bc000 r--p 00158000 08:02 81930
7fa6e72bc000-7fa6e72be000 rw-p 0015b000 08:02 81930
7fa6e72be000-7fa6e72c3000 rw-p 7fa6e72be000 00:00 0
7fa6e72c3000-7fa6e72c5000 r-xp 00000000 08:02 81936
7fa6e72c5000-7fa6e74c5000 ---p 00002000 08:02 81936
7fa6e74c5000-7fa6e74c7000 rw-p 00002000 08:02 81936
7fa6e74c7000-7fa6e74fe000 r-xp 00000000 08:02 82217
7fa6e74fe000-7fa6e76fd000 ---p 00037000 08:02 82217
7fa6e76fd000-7fa6e7702000 rw-p 00036000 08:02 82217
7fa6e7702000-7fa6e771f000 r-xp 00000000 08:02 81927
7fa6e771f000-7fa6e77e9000 r--p 00000000 08:02 271736
```

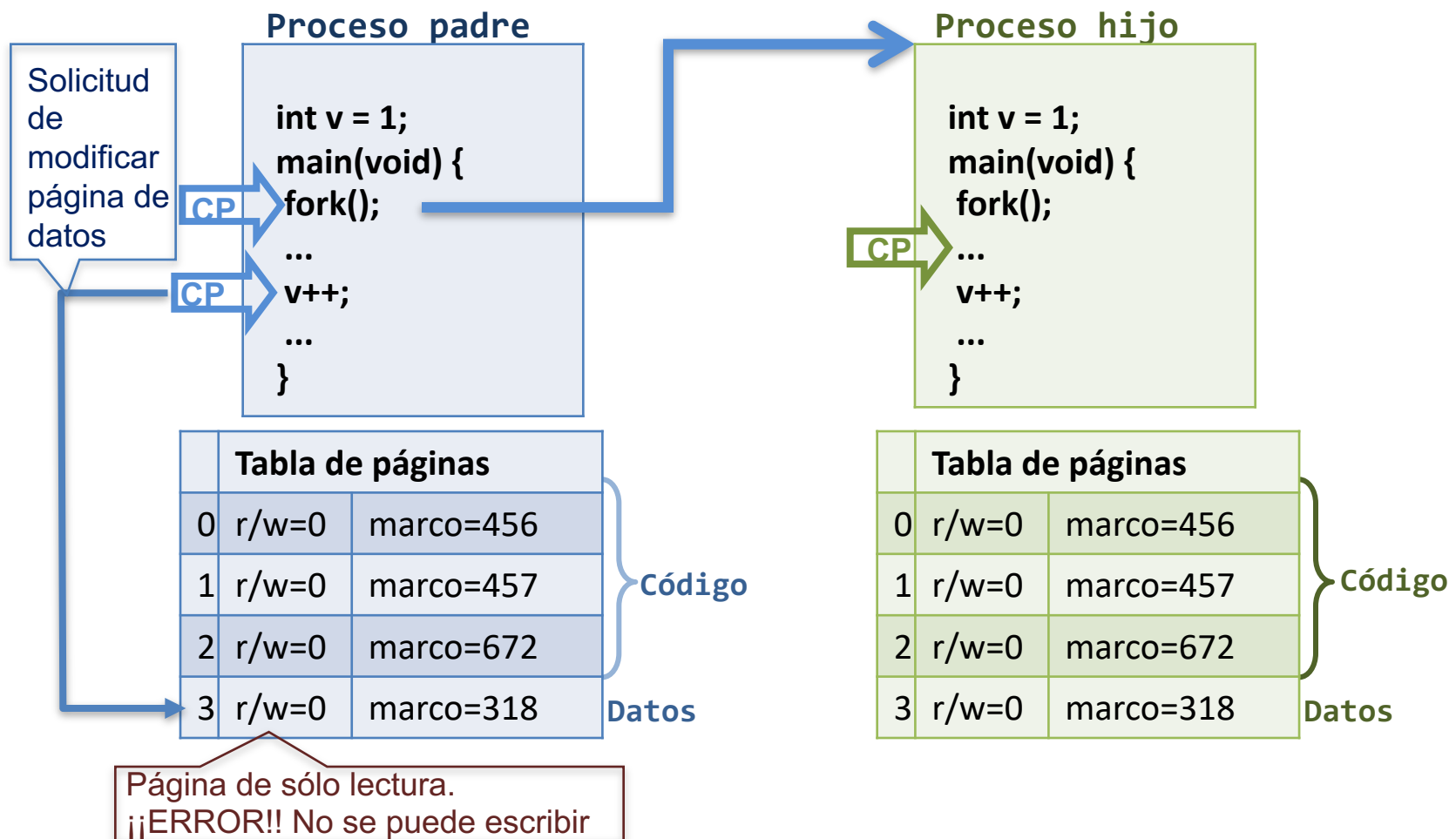
```
/bin/bash
/bin/bash
[heap]
/lib/libnss_files-2.7.so
/lib/libnss_files-2.7.so
/lib/libnss_files-2.7.so
/lib/libnss_nis-2.7.so
/lib/libnss_nis-2.7.so
/lib/libnss_nis-2.7.so
/lib/libnsl-2.7.so
/lib/libnsl-2.7.so
/lib/libnsl-2.7.so
/lib/libnss_compat-2.7.so
/lib/libnss_compat-2.7.so
/lib/libnss_compat-2.7.so
/lib/libc-2.7.so
/lib/libc-2.7.so
/lib/libc-2.7.so
/lib/libc-2.7.so
/lib/libdl-2.7.so
/lib/libdl-2.7.so
/lib/libdl-2.7.so
/lib/libncurses.so.5.6
/lib/libncurses.so.5.6
/lib/libncurses.so.5.6
/lib/libd-2.7.so
/usr/share/locale-langpack/es/LC_MESSAGES/bash.mo
```

```
7fa6e77e9000-7fa6e7828000 r--p 00000000 08:02 439402
7fa6e7828000-7fa6e7909000 r--p 00000000 08:02 439411
7fa6e7909000-7fa6e790b000 rw-p 7fa6e7909000 00:00 0
7fa6e790b000-7fa6e790c000 r--p 00000000 08:02 439403
7fa6e790c000-7fa6e790d000 r--p 00000000 08:02 28628
7fa6e790d000-7fa6e790e000 r--p 00000000 08:02 28629
7fa6e790e000-7fa6e790f000 r--p 00000000 08:02 21591
/usr/lib/locale/es_ES.utf8/LC_MESSAGES/SYS_LC_MESSAGES
7fa6e790f000-7fa6e7910000 r--p 00000000 08:02 439406
7fa6e7910000-7fa6e7911000 r--p 00000000 08:02 439410
7fa6e7911000-7fa6e7912000 r--p 00000000 08:02 28631
7fa6e7912000-7fa6e7913000 r--p 00000000 08:02 28633
7fa6e7913000-7fa6e7914000 r--p 00000000 08:02 439407
7fa6e7914000-7fa6e791b000 r--s 00000000 08:02 446759
7fa6e791b000-7fa6e791c000 r--p 00000000 08:02 28635
7fa6e791c000-7fa6e791f000 rw-p 7fa6e791c000 00:00 0
7fa6e791f000-7fa6e7921000 rw-p 0001d000 08:02 81927
7fff8bc0f000-7fff8bc24000 rw-p 7fffffe9000 00:00 0
7fff8bd1d000-7fff8bd1f000 r-xp 7fff8bd1d000 00:00 0
ffffffff600000-ffffffff601000 r-xp 00000000 00:00 0
gandreu@shell-sisop:~$
/usr/lib/locale/es_ES.utf8/LC_CTYPE
/usr/lib/locale/es_ES.utf8/LC_COLLATE
/usr/lib/locale/es_ES.utf8/LC_NUMERIC
/usr/lib/locale/es_ES.utf8/LC_TIME
/usr/lib/locale/es_ES.utf8/LC_MONETARY
/usr/lib/locale/es_ES.utf8/LC_PAPER
/usr/lib/locale/es_ES.utf8/LC_NAME
/usr/lib/locale/es_ES.utf8/LC_ADDRESS
/usr/lib/locale/es_ES.utf8/LC_TELEPHONE
/usr/lib/locale/es_ES.utf8/LC_MEASUREMENT
/usr/lib/gconv/gconv-modules.cache
/usr/lib/locale/es_ES.utf8/LC_IDENTIFICATION
/lib/ld-2.7.so
[stack]
[vdso]
[vsyscall]
```


- **Copia al escribir “Copy-on-Write”**
 - Técnica de LINUX para **realizar eficientemente la copia de páginas de memoria**, con ahorro memoria y tiempo
 - Al crear un proceso, padre e hijo comparten página de datos y pila en memoria
 - Todos las páginas que comparten se marcan como **“read-only”**
 - Si se intenta acceder a una página marcada como **“read-only”** para modificarla la MMU produce un fallo en el acceso, entonces:
 - El s.o. realiza una copia de dicha página con permisos de escritura para el proceso que quiere escribir
 - » Si hay más de dos procesos, el resto sigue sin poder escribir
 - » Si sólo queda un proceso utilizando la página podrá realizar escrituras, modificando previamente el bit en su descriptor
 - Se reinicia de nuevo la instrucción que produjo el fallo
 - **Ventaja:** LINUX evita copiar páginas en memoria que no se utilizan ahorrando la memoria correspondiente y el tiempo de copiarlas

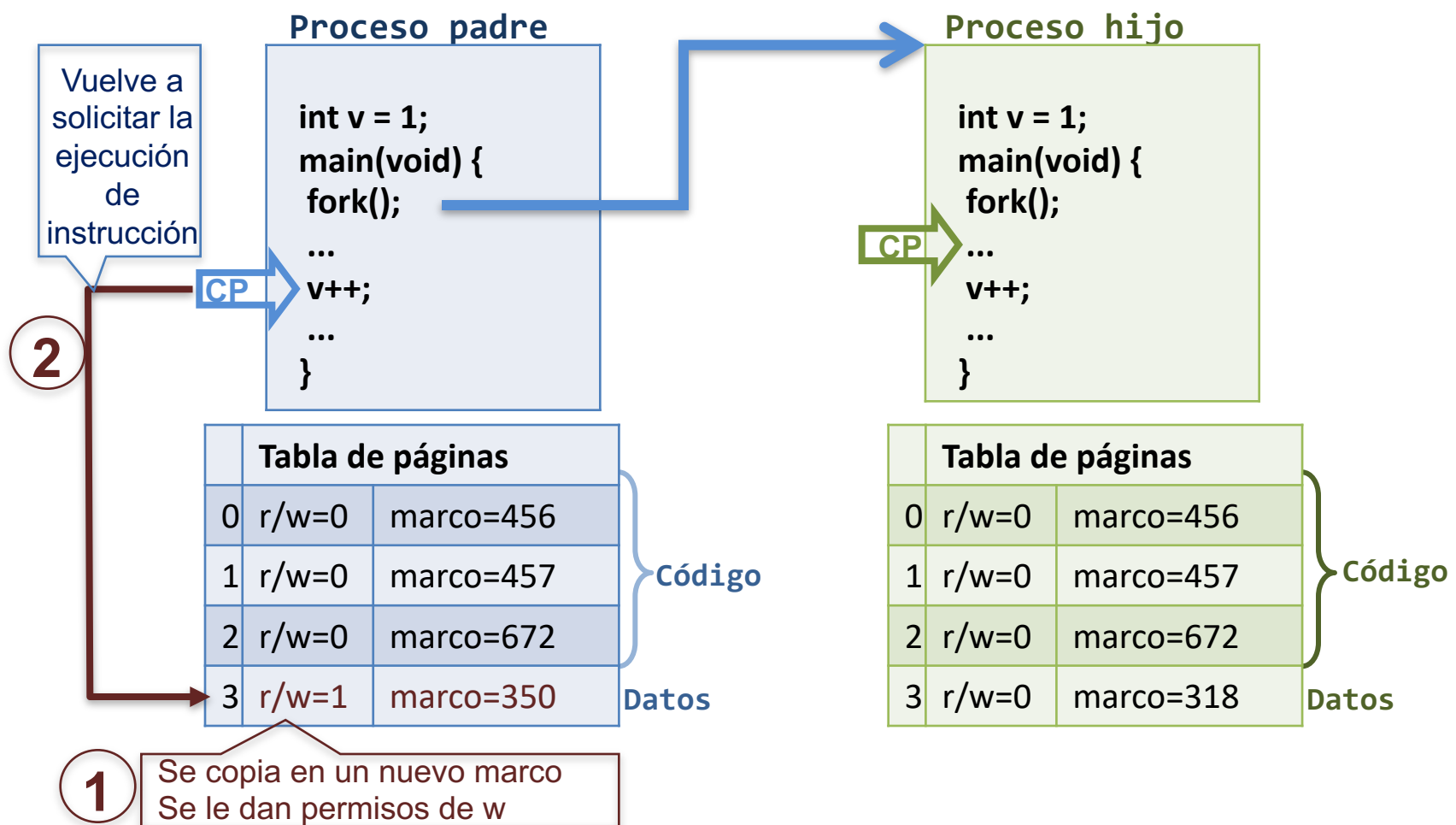
• Llamada `fork()` “Copy-on-Write”

- Al crear un proceso, padre e hijo comparten página de datos y pila en memoria
 - Todas las páginas que comparten se marcan como “**read-only**”
 - Si se intenta acceder a una página marcada como “read-only” para modificarla la MMU produce un fallo en el acceso, entonces:



• Llamada *fork()* “Copy-on-Write”

- Al crear un proceso, padre e hijo comparten página de datos y pila en memoria
 - Todas las páginas que comparten se marcan como “**read-only**”
 - Si se intenta acceder a una página marcada como “read-only” para modificarla la MMU produce un fallo en el acceso, entonces:



- Introducción
- Mapa de memoria de un proceso Linux
- **Archivos mapeados en memoria**
- Bibliotecas de enlace dinámico

- **Archivos mapeados o proyectados en memoria**

- El contenido del archivo o parte de éste se incluye en el mapa de memoria del proceso

`mmap()` en POSIX

`CreateFileMapping()` en Win32

- Ventajas:

- **Mejora el tiempo de acceso** al contenido del archivo. Se accede al contenido del archivo como si fuera una estructura de datos en memoria
- **Se evitan copias intermedias** de la información. El S.O. transfiere directamente la información entre la región de memoria y el archivo

- NO requiere instrucciones de E/S (*read*, *write*) para acceder al archivo

Se accede al contenido con instrucciones y variables propias de los programas:

```
for (i=0; i<bstat.st_size; i++)  
    if (*p++==caracter) contador++;
```

p es un puntero al carácter del archivo

*p es el contenido de un puntero a carácter y por tanto un carácter

- **mmap**: llamada al sistema POSIX que mapea archivos en memoria

```
caddr_t mmap(caddr_t direc, size_t longitud,  
             int protec, int indicador, int fd, off_t despl)
```

- **mmap** crea una nueva región en el mapa de memoria del proceso que la invoca. Se puede especificar si es privada o compartida y el tipo de protección
 - **direc**: indica la dirección del mapa donde se quiere que se proyecte el fichero. Si se pone un 0 la decisión la toma el sistema. **mmap** siempre devuelve la dirección de proyección utilizada
 - **fd**: descriptor del fichero que se quiere proyectar (previamente abierto)
 - **despl y longitud** establecen qué zona del fichero se proyecta, desde **despl** hasta **despl+longitud-1**
 - **protec**: PROT_READ, PROT_WRITE, PROT_EXEC y combinaciones
 - **indicador**: MAP_SHARED, MAP_PRIVATE

munmap: elimina una proyección previa o parte de ésta

ejemplo1.c : programa que mapea un archivo en memoria y cuenta el número de veces que aparece un determinado carácter. El nombre del archivo y el carácter se pasan como argumentos en la línea de órdenes

```
$ejemplo1 c ejemplo1.c
```

Archivo a mapear y donde se busca el carácter "c"

```
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/mman.h>
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>

int main(int argc, char *argv[] ) {
    int i,fd,contador;
    char *p,*org;
    struct stat bstat;
    char caracter;
    caracter= argv[1][0];
    fd=open(argv[2], O_RDONLY); /* Abre fichero */
    fstat(fd, &bstat); /* Averigua long. fichero */
    /* Se proyecta el fichero */
    org=mmap((caddr_t) 0, bstat.st_size, PROT_READ,MAP_SHARED, fd, 0);
    close(fd); /* Se cierra el fichero */
    /* Bucle de acceso */
    p=org;
    contador=0;
    for (i=0; i<bstat.st_size; i++)
        if (*p++==caracter) contador++;
    /* Se elimina la proyeccion */
    munmap(org, bstat.st_size);
    printf("%d\n", contador);
}
```

Mapa de memoria del proceso “ejemplo1”

antes de proyectar archivo

```
pblanes$ pmap 21914:
21914  ./ejemplo1 c ejemplo1.c
08048000  4K r-x-- /home/naomac/fso/ejemplo1
08049000  4K rw--- /home/naomac/fso/ejemplo1
b7de1000  4K rw--- [ anon ]
b7de2000  1316K r-x-- /lib/tls/i686/cmov/libc-2.7.so
b7f2b000  4K r---- /lib/tls/i686/cmov/libc-2.7.so
b7f2c000  8K rw--- /lib/tls/i686/cmov/libc-2.7.so
b7f2e000  12K rw--- [ anon ]
b7f42000  12K rw--- [ anon ]
b7f45000  4K r-x-- [ anon ]
b7f46000  104K r-x-- /lib/ld-2.7.so
b7f60000  8K rw--- /lib/ld-2.7.so
bf986000  84K rw--- [ stack ]
total 1564K
```

una vez proyectado el archivo

```
pblanes$ pmap 21914
21914: ./ejemplo1 c ejemplo1.c
08048000  4K r-x-- /home/naomac/fso/ejemplo1
08049000  4K rw--- /home/naomac/fso/ejemplo1
b7de1000  4K rw--- [ anon ]
b7de2000  1316K r-x-- /lib/tls/i686/cmov/libc-2.7.so
b7f2b000  4K r---- /lib/tls/i686/cmov/libc-2.7.so
b7f2c000  8K rw--- /lib/tls/i686/cmov/libc-2.7.so
b7f2e000  12K rw--- [ anon ]
b7f41000  4K r--s- /home/naomac/fso/ejemplo1.c
b7f42000  12K rw--- [ anon ]
b7f45000  4K r-x-- [ anon ]
b7f46000  104K r-x-- /lib/ld-2.7.so
b7f60000  8K rw--- /lib/ld-2.7.so
bf986000  84K rw--- [ stack ]
total 1568K
```

archivo mapeado ejemplo1.c

“**mapear.c**”: programa que muestra el mapa de memoria del proceso antes y después de mapear en memoria el archivo que se le pasa como argumento

Para compilar: **\$gcc mapear.c -o mapear**

Para ejecutar: **\$mapear mapear.c**

```
#int main (int argc, char *argv[])
{
    int fd;
    void *mapeo;
    struct stat stadbuf;
    char path_maps[80];

    //Abrir el archivo a ser mapeado
    if (argc!=2) {
        puts("Usar: mapear NombreArchivo\n");
        exit(EXIT_FAILURE) ;
    }

    if ((fd=open(argv[1],O_RDONLY))<0)
        error("Fallo en la apertura (open)\n");
    //Obtener la longitud del archivo a mapear
    fstat(fd, &stadbuf);
    //fstat vuelca su informacion en estadobuf

    //MOSTRAR MAPA
    printf(" MAPA MEMORIA PROCESO /proc/%d/maps \n", getpid());
    construye_orden(path_maps);
    system(path_maps); //Llamada al sistema ejecutar orden
```

```
#include <sys/types.h>
#include <sys/mman.h>
#include <sys/stat.h>
#include <unistd.h>
#include <fcntl.h>
#include <stdlib.h>
#include <stdio.h>

void error (char * mensaje)
{
    perror(mensaje);
    exit(EXIT_FAILURE);
}

void construye_orden(char orden[80])
{
    //Construye orden para mostrar MAPA
    sprintf(orden, "cat /proc/%d/maps", getpid());
}
```

continua ...

continuación de “mapear.c.....”

```
//Mapear el archivo de entrada
if ((mapeo=mmap(0,stadbuf.st_size,PROT_READ,MAP_SHARED,fd,0))== MAP_FAILED)
error("Fallo al mapear (open)");
close(fd); //cierro fichero

//MOSTRAR MAPA
printf ("\n\n FICHERO MAPEADO EN MEMORIA\n");
system(path_maps);//Llamada al sistema para ejecutar orden

munmap(mapeo,stadbuf.st_size); //Elimino mapeo

printf ("\n\n ELIMINADO EL MAPEO DEL FICHERO EN MEMORIA\n");

system(path_maps);

exit(EXIT_SUCCESS);

}/**Fin de main**/
```

Se realiza una llamada para ejecutar la orden
“cat /proc/%d/maps” creada y mostrar el
archivo maps de este proceso

- Introducción
- Mapa de memoria en Linux
- Archivos mapeados en memoria
- **Bibliotecas de enlace dinámico**

- **Bibliotecas o librerías**

- Son archivos binarios que contienen código de funciones o subprogramas
- Existen dos formas de enlazarlas al programa:
 - **Con enlace estático:** el ejecutable incluye todo el código
 - Archivo ejecutable = Código del programa + código de funciones y librerías
 - archivos con extensión **.lib de Win32, .a de UNIX/LINUX**
 - **Enlace dinámico:** el archivo ejecutable contiene las referencias a las rutinas de biblioteca a utilizar y al área de memoria que necesitara para almacenarlas
 - Su carga en memoria se realiza a la invocación de las mismas mediante el mapeo de las librerías en memoria. Esta labor la realiza un programa cargador del librerías mediante la llamada **dlopen**
 - Son los archivos **.dll de Win32 o .lib de UNIX/LINUX**

- **Enlace estático**

- Desventajas
 - Ejecutables grandes
 - Código de función de biblioteca repetido en muchos ejecutables
 - Múltiples copias en memoria del código de función de biblioteca
 - Actualización de biblioteca implica volver a montar
- Ventajas
 - El ejecutable contiene todo el código que necesita para su ejecución

- **Enlace dinámico**

- Ventajas
 - Menor tamaño ejecutables, ahorro de espacio en disco y en memoria
 - Código de rutinas de biblioteca sólo en fichero de biblioteca
 - Procesos pueden compartir código de biblioteca
 - Actualización automática de bibliotecas: Uso de versiones
- Desventajas
 - En tiempo de ejecución se resuelve el enlace, esto puede ocasionar retrasos temporales en la ejecución

- Programa ejemplo1
 - Ejecutable generado **con enlace estático** de la librería math

```
pblanes$ gcc ejemplo1.c -static -o ejemplo1 -lm
pblanes$ ls -l
total 660
-rwxr-xr-x 1 pblanes disca-upvnet 670227 2011-10-20 15:54 ejemplo1
-rw-r--r-- 1 pblanes disca-upvnet 905 2011-10-20 13:30 ejemplo1.c
```

- Ejecutable generado **con enlace dinámico** de la librería math

```
pblanes$ gcc ejemplo1.c -o ejemplo1 -lm
pblanes$ ls -l
total 16
-rwxr-xr-x 1 pblanes disca-upvnet 10301 2011-10-20 15:56 ejemplo1
-rw-r--r-- 1 pblanes disca-upvnet 905 2011-10-20 13:30 ejemplo1.c
```