

Fundamentos de els Sistemes Operatius (FSO)

Departamento de Informàtica de Sistemes i Computadores (DISCA)

Universitat Politècnica de València

Bloc Temàtic 4: Gestió de Memòria

Seminari 9

Mapa de memòria d'un procés

Linux

fSO

DISCA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

- **Objectius**

- Comprendre el **concepte de mapa de memòria d'un procés**
- Coeixer les característiques del mapa de memòria en procés Linux
- Descriure la tècnica per a **mapejar arxius en memòria**
- Diferenciar els avantatges i inconvenients de treballar amb **biblioteques estàtiques i dinàmiques**

- **Bibliografia**

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

- **Introdució**
- Mapa de memòria d'un procés en LINUX
- Arxius mapejats en memòria
- Biblioteques d'enllaç dinàmic

- **Mapa de memòria d'un procés**

- El S. O. gestiona el mapa de memòria d'un procés durant la seua vida
- El mapa de memòria és un **atribut** del procés
- Conté informació sobre les distintes àrees del procés:
 - Codi, dades, pila ...

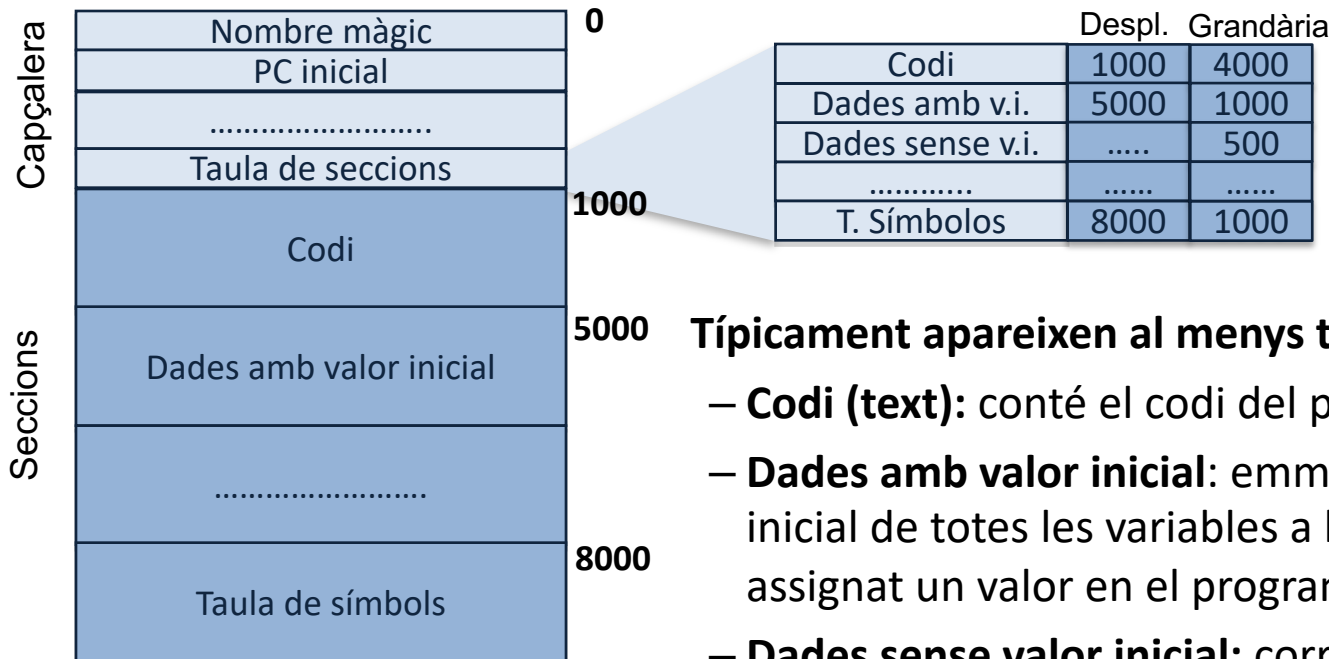
PCB



- **Mapa de memòria inicial d'un procés:** està molt vinculat amb l'arxiu que conté el codi **executable** d'aquest procés
- Actualment els SO ofereixen un **model de memòria dinàmic** mitjançant regions que donen suport a aquestes àrees, de manera que durant l'execució del procés
 - Es poden **crear noves regions** que no obtenen les dades del fitxer executable: pila, arxius projectats, memòria dinàmica
 - Es poden **eliminar regions**

• Format simplificat d'un arxiu executable

- El resultat final de compilar i enllaçar un programa es generar un arxiu executable que continga el codi màquina del programa
- Un executable està estructurat **en capçalera i un conjunt de seccions**
 - **Capçalera:** conté informació de control que permet interpretar la resta del contingut del fitxer executable
 - **Seccions:** Cada executable te un conjunt de seccions

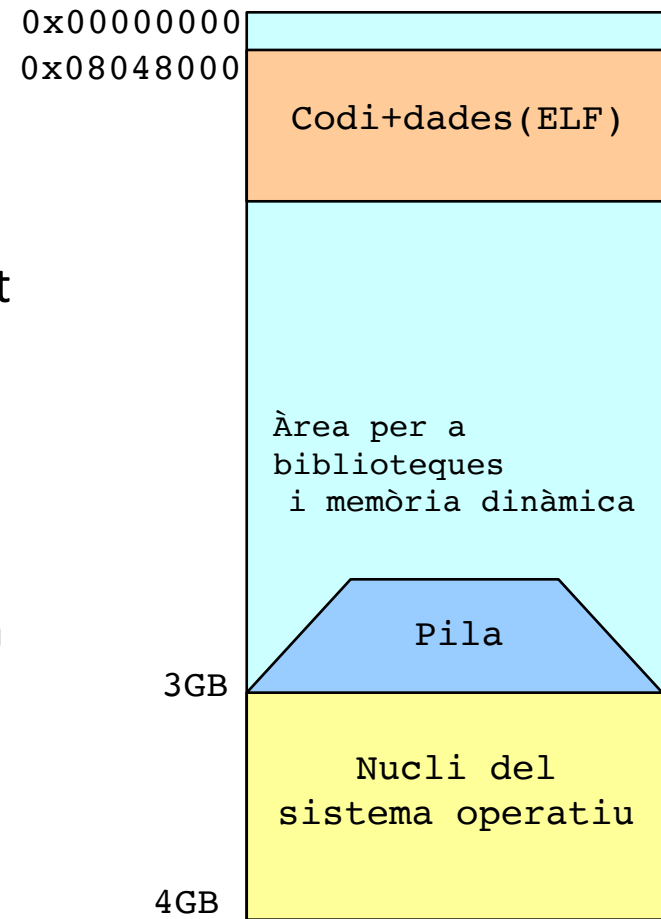


Típicament apareixen al menys tres seccions:

- **Codi (text):** conté el codi del programa
- **Dades amb valor inicial:** emmagatzema el valor inicial de totes les variables a les quals se li ha assignat un valor en el programa
- **Dades sense valor inicial:** correspon a variables a les quals no se li ha assignat un valor inicial

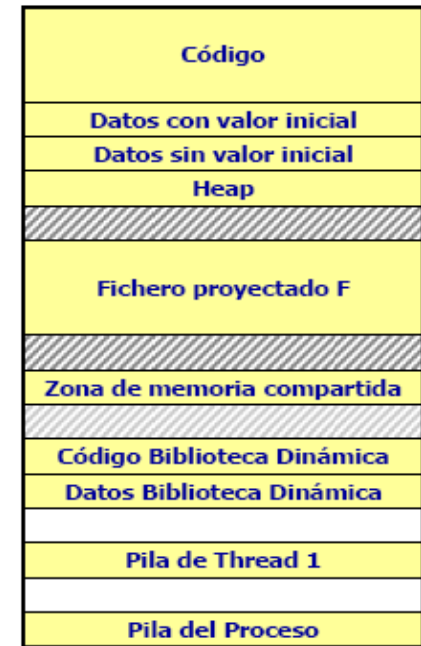
- Introdució
- **Mapa de memòria d'un procés Linux**
- Arxius mapejats en memòria
- Biblioteques d'enllaç dinàmic

- Mapa de memòria Linux en **versions de LINUX 2.6.x** sobre arquitectura de 32 bits:
 - Espai lògic de 4GB (2^{32})
 - 1er Gbyte: Codi + dades inicialitzades que comencen en l'adreça 0x08048000 (format ELF)
 - 3er Gbyte per a pila: la pila creix cap a posicions decreixents i comença en la part alta del 3er GB
 - 4t Gbyte: les adreces més altes es reserven per al S.O.
 - La resta de l'espai pot utilitzar-se per a dades i biblioteques d'enllaç dinàmic

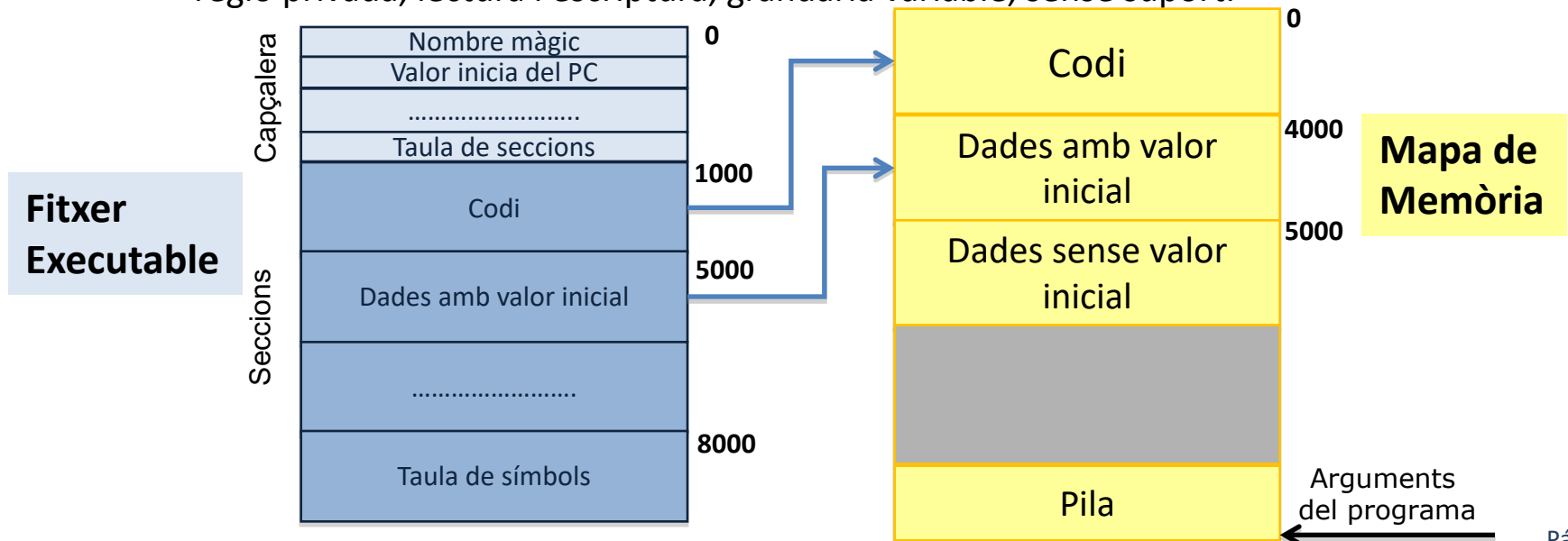


- **Mapa de memòria esta format per regions**

- Una regió té associada una determinada informació
 - codi, dades amb valor inicial, dades sense valor inicial, fitxers projectats, codis biblioteques dinàmiques
- Cada regió és una **zona contígua** caracteritzada per l'adreça on comença en del mapa i la seua grandària
- Característiques de les regions
 - **Suport:** on s'emmagatzema la informació de la regió
 - **en arxiu:** el contingut inicial de la regió està en un fitxer
 - **sense suport:** la regió no té contingut inicial
 - Tipus de **compartició:**
 - **Privada (p)** : el contingut de la regió només es accessible pel procés
 - **Compartida:** el contingut de la regió pot ser accessible per diversos processos
 - **Protecció:** Tipus d'accés permés a eixa regió
 - lectura, escriptura i/o execució
 - **Grandària:** fixa o variable



- En executar un programa es crea un mapa de memòria inicial a partir de l'arxiu executable
- **Cada secció de l'executable** dóna lloc a **una regió del mapa inicial**
 - **Codi (text):**
 - regió compartida, lectura i execució, grandària fixa, suport en arxiu executable
 - **Dades amb valor inicial:** cada procés necessita una còpia pròpia de les variables.
 - regió privada, lectura i escriptura, grandària fixa, suport en arxiu executable
 - **Dades sense valor inicial:** en alguns llenguatges s'ompli a zero
 - regió privada, lectura i escriptura, grandària fixa, sense suport
 - **Pila:** Creix cap a adreces baixes. Inicialment només conté els arguments de la crida al programa
 - regió privada, lectura i escriptura, grandària variable, sense suport.



- **El mapa de memòria d'un procés és dinàmic.** Durant l'execució del procés es poden crear regions com:
 - **Regió de Heap**
 - Suport de memòria dinàmica (`malloc` en C)
 - Privada, lectura i escriptura, grandària variable, sense suport (inicialitzada a zero)
 - Creix cap a adreces més altes
 - **Arxiu projectat**
 - Quan es mapeja un fitxer es crea una regió associada a l'arxiu projectat
 - Grandària variable, suport en arxiu
 - Protecció i caràcter compartit o privat especificat en la projecció
 - **Memòria compartida**
 - Regió associada a la zona de memòria compartida (per a la comunicació entre processos)
 - Compartida, grandària variable, sense suport (inicialitzada a 0)
 - Protecció especificada pel programa
 - **Piles de Fils (threads)**
 - Cada pila de fil es correspon amb una regió.
 - Aquestes regions consten de les mateixes característiques que les associades a la pila del procés

- Disposem de dues opcions a l'hora de **visualitzar el mapa de memòria d'un procés en Linux**:

- Llistar l'arxiu **maps** del procés ubicat en */proc/PID_del_proces*

- **\$cat /proc/PID/maps**

- » Rang d'adreces lògiques
- » Permissos
- » Desplaçament des de l'inici del fitxer executable
- » Dispositiu
- » Node-i
- » Nom del fitxer projectat

- Executar l'ordre del shell **\$pmap PID**

- » Adreça lògica base
- » Grandària
- » Permissos
- » Nom fitxer projectat

Nota: Substituint el PID per la variable **\$\$** se fa referència al **procés** que està en **execució**. Exemple:

```
$ pmap $$
```

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

Exemple: ordre \$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

Exemple: ordre \$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	----	00000000000016000	008:00002	libnsl-2.7.so
00007fa6e6d54000	8	rw---	00000000000015000	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:~\$

• Arxiu `/proc/PID/maps`

- Conté les regions de memòria actualment associades al procés amb identificador PID i els seus permissos d'accés
- El format de l'arxiu *maps* és:

Adreça	permissos	desplaz.	disp	node-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	rxp	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	rxp	00000000	00:00	0	

Adreça: Espai d'adreces que ocupa el procés

Permisos:

r = leer
w = escriure
x = executar
s = compartit
p = privat (còpia en escriptura)

Node-i: Node-i del dispositiu, 0 indica que no hi ha node-i associat

Dispositiu: És el dispositiu (major number : minor number)

Desplaçament: Es el desplaçament dins de l'arxiu que el suporta

Exemple: arxiu 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]

Exemple: arxiu 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
```

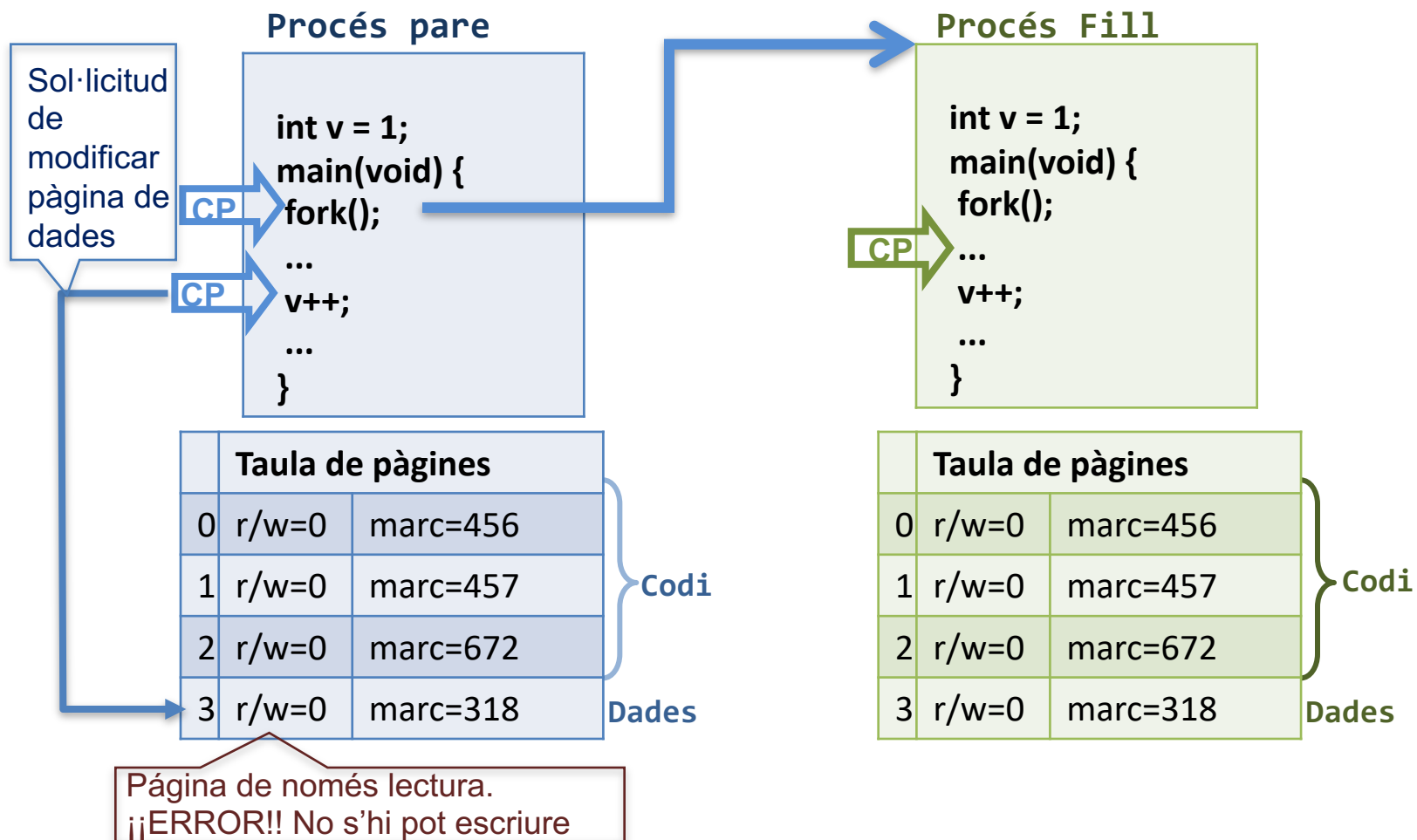
```
/bin/bash
/bin/bash
[heap]
/lib/libnss_files-2.7.so
/lib/libnss_files-2.7.so
/lib/libnss_files-2.7.so
/lib/libnss_nis-2.7.so
/lib/libnss_nis-2.7.so
/lib/libnss_nis-2.7.so
/lib/libnsl-2.7.so
/lib/libnsl-2.7.so
/lib/libnsl-2.7.so
/lib/libnss_compat-2.7.so
/lib/libnss_compat-2.7.so
/lib/libnss_compat-2.7.so
/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 /usr/lib/locale/es_ES.utf8/LC_CTYPE
7fa6e7828000-7fa6e7909000 r--p 00000000 08:02 439411 /usr/lib/locale/es_ES.utf8/LC_COLLATE
7fa6e7909000-7fa6e790b000 rw-p 7fa6e7909000 00:00 0
7fa6e790b000-7fa6e790c000 r--p 00000000 08:02 439403 /usr/lib/locale/es_ES.utf8/LC_NUMERIC
7fa6e790c000-7fa6e790d000 r--p 00000000 08:02 28628 /usr/lib/locale/es_ES.utf8/LC_TIME
7fa6e790d000-7fa6e790e000 r--p 00000000 08:02 28629 /usr/lib/locale/es_ES.utf8/LC_MONETARY
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 /usr/lib/locale/es_ES.utf8/LC_PAPER
7fa6e7910000-7fa6e7911000 r--p 00000000 08:02 439410 /usr/lib/locale/es_ES.utf8/LC_NAME
7fa6e7911000-7fa6e7912000 r--p 00000000 08:02 28631 /usr/lib/locale/es_ES.utf8/LC_ADDRESS
7fa6e7912000-7fa6e7913000 r--p 00000000 08:02 28633 /usr/lib/locale/es_ES.utf8/LC_TELEPHONE
7fa6e7913000-7fa6e7914000 r--p 00000000 08:02 439407 /usr/lib/locale/es_ES.utf8/LC_MEASUREMENT
7fa6e7914000-7fa6e791b000 r--s 00000000 08:02 446759 /usr/lib/gconv/gconv-modules.cache
7fa6e791b000-7fa6e791c000 r--p 00000000 08:02 28635 /usr/lib/locale/es_ES.utf8/LC_IDENTIFICATION
7fa6e791c000-7fa6e791f000 rw-p 7fa6e791c000 00:00 0
7fa6e791f000-7fa6e7921000 rw-p 0001d000 08:02 81927 /lib/ld-2.7.so
7fff8bc0f000-7fff8bc24000 rw-p 7fffffe9000 00:00 0 [stack]
7fff8bd1d000-7fff8bd1f000 r-xp 7fff8bd1d000 00:00 0 [vdso]
fffffffff600000-fffffffff601000 r-xp 00000000 00:00 0 [vsyscall]
gandreu@shell-sisop:~$
```


- **Còpia en escriure “Copy-on-Write”**
 - Tècnica de LINUX per a **fa eficientment la còpia de pàgines de memòria**, amb estalvi de memòria i de temps
 - En crear un procés, pare i fill comparteixen les pàgines de dades i de pila en memòria
 - Totes les pàgines compartides es marquen com **“read-only”**
 - Si s'intenta accedir a una pàgina marcada com **“read-only”** per a modificar-la, la MMU produeix una fallada en l'accés, aleshores:
 - El s.o. fa una còpia d'aquesta pàgina amb permissos d'escriptura per al procés que hi vol escriure
 - » Si hi ha més de dos processos, la resta continua sense poder escriure
 - » Si només queda un procés utilitzant la pàgina podrà fer escriptures, modificant prèviament el bit en el seu descriptor
 - Es reinicia de nou la instrucció que va produir la fallada
 - **Avantatge:** Linux evita copiar les pàgines de memòria que no s'utilitzen estalviant la memòria corresponent i el temps de copiar-les

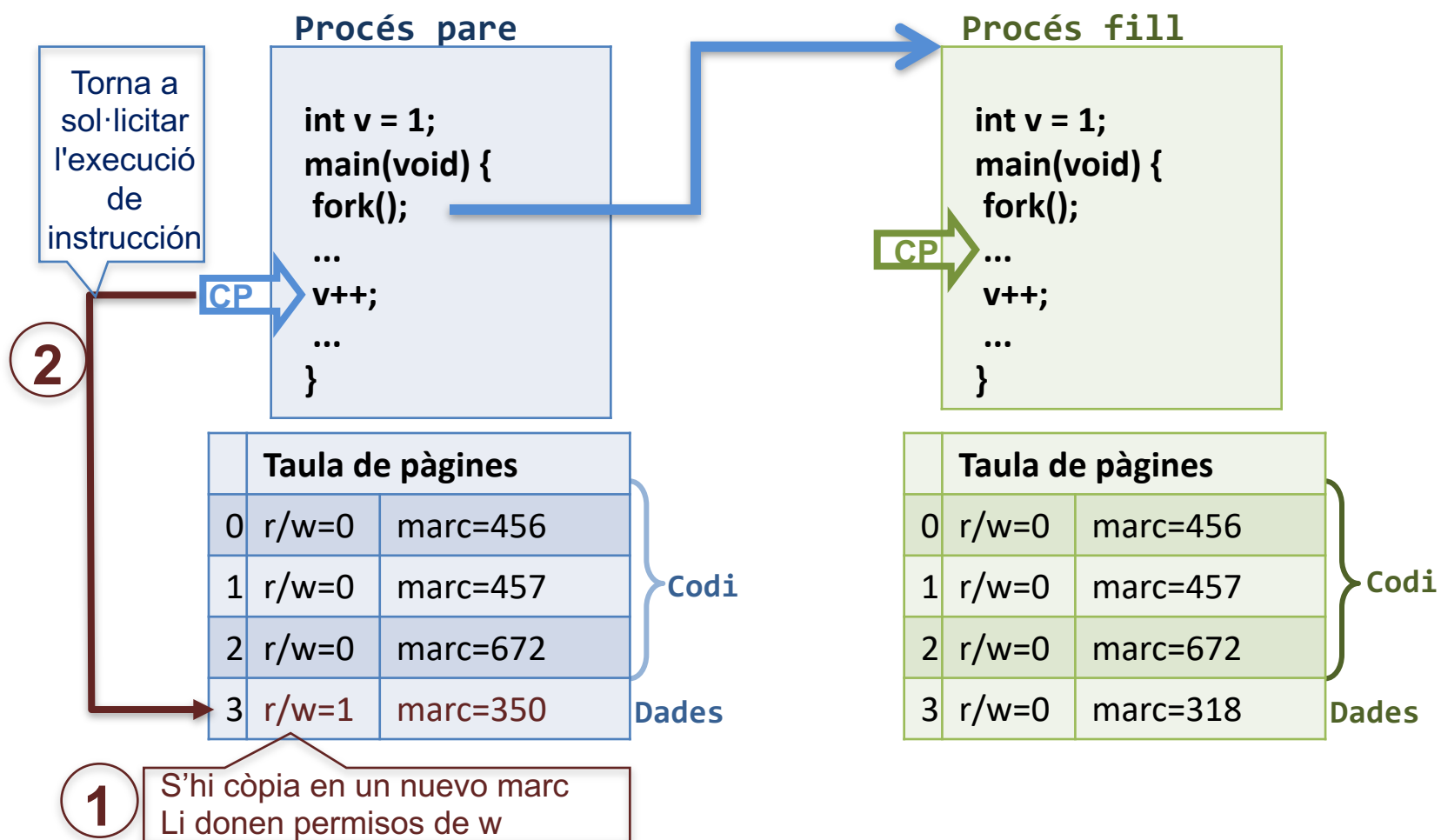
- Crida *fork()* “Copy-on-Write”

- En crear un procés, pare i fill comparteixen pàgina de dades i pila en memòria
 - Totes les pàgines que comparteixen se marquen como “read-only”
 - Si s'intenta accedir a una pàgina marcada com a “read-only” per a modificar-la la MMU produeix una fallada en l' accés, aleshores:



- Crida *fork()* “Copy-on-Write”

- Al crear un procés, pare i fill comparteixen pàgina de dades i pila en memòria
 - Totes les pàgines que comparteixen se marquen como “read-only”
 - Si s'intenta accedir a una pàgina marcada com a “read-only” per a modificar-la la MMU produeix una fallada en l' accés, aleshores:



- Introdució
- Mapa de memòria d'un procés Linux
- **Arxius mapejats en memòria**
- Bibliotecas d'enllaç dinàmic

- **Arxius mapejats o projectats en memòria**

- El contingut de l'arxiu o part d'aquest s'inclou en el mapa de memòria del procés

mmap() en POSIX

CreateFileMapping() en Win32

- Avantatges:

- **Millora el temps de accés** al contingut de l'arxiu. S'hi accedeix si fóra una estructura de dades en memòria
- **S'eviten còpies intermitges** de la informació. El S.O. transfereix directament la informació entre la regió de memòria i l'arxiu

- NO requereix instruccions d'E/S (*read*, *write*) per a accedir a l'arxiu

S'accedeix al contingut amb instruccions i variables pròpies dels programes:

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

P és un punter al caràcter de l'arxiu

*p és el contingut d'un punter a caràcter i per tant un caràcter

- **mmap**: crida al sistema POSIX que mapeja arxius en memòria

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

- **mmap** crea una nova regió en el mapa de memòria del procés que la invoca. Es pot especificar si és privada o compartida i el tipus de protecció
 - **direc**: indica l'adreça del mapa on es vol que es projecte el fitxer. Si es fica un 0 la decisió la pren el sistema. **mmap** sempre torna l'adreça de projecció utilitzada
 - **fd**: descriptor del fitxer que es vol projectar (prèviament obert)
 - **despl i longitud** estableixen quina zona del fitxer es projecta, des de *despl* fins *despl+longitud-1*
 - **protec**: PROT_READ, PROT_WRITE, PROT_EXEC i combinacions
 - **indicador**: MAP_SHARED, MAP_PRIVATE

munmap: elimina una projecció prèvia o part d'aquesta

exemple1.c : programa que mapeja un arxiu en memòria i compta el nombre de vegades que apareix un determinat caràcter. El nom de l'arxiu i el caràcter es passen com a arguments en la línia d'ordres

```
$exemple1 c exemple1.c
```

Arxiu a mapejar i on es busca el caràcter "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,comptador;
    char *p,*org;
    struct stat bstat;
    char caracter;
    caracter= argv[1][0];
    fd=open(argv[2], O_RDONLY); /* Obri fitxer */
    fstat(fd, &bstat); /* Averigua long. fitxer */
    /* Es projecta el fitxer */
    org=mmap((caddr_t) 0, bstat.st_size, PROT_READ,MAP_SHARED, fd, 0);
    close(fd); /* Es tanca el fitxer */
    /* Bucle d'accés */
    p=org;
    comptador=0;
    for (i=0; i<bstat.st_size; i++)
        if (*p++==caracter) comptador++;
    /* S'elimina la projecció */
    munmap(org, bstat.st_size);
    printf("%d\n", comptador);
}
```

Mapa de memòria del procés “exemple1”

Abans de projectar arxiu

```
pblanes$ pmap 21914:
21914  ./exemple1 c exemple1.c
08048000  4K r-x-- /home/naomac/fso/exemple1
08049000  4K rw--- /home/naomac/fso/exemple1
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 vegada projectat l'arxiu

```
pblanes$ pmap 21914
21914:  ./exemple1 c exemple1.c
08048000  4K r-x-- /home/naomac/fso/exemple1
08049000  4K rw--- /home/naomac/fso/exemple1
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/exemple1.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
```

arxiu mapejat exemple1.c

“**mapejar.c**” : programa que mostra el mapa de memòria del procés abans i després de mapejar en memòria l'arxiu que se li passa com a argument

Per a compilar: **\$gcc mapejar.c -o mapejar**

Per a executar: **\$mapejar mapejar.c**

```
#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 ordre[80])
{
    //Construeix ordre per a mostrar MAPA
    sprintf(ordre, "cat /proc/%d/maps", getpid());
}

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

    //Abrir l'arxiu a ser mapeado
    if (argc!=2) {
        puts("Usar: mapejar NombreArchivo\n");
        exit(EXIT_FAILURE) ;
    }

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

    //MOSTRAR MAPA
    printf(" MAPA MEMORIA PROCES /proc/%d/maps \n", getpid());
    construye_orden(path_maps);
    system(path_maps); //Crida al sistema ejecutar ordre
}
```

```
#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 ordre[80])
{
    //Construeix ordre per a mostrar MAPA
    sprintf(ordre, "cat /proc/%d/maps", getpid());
}
```

continua ...

Continuació de “mapejar.c.....”

```
//Mapejar l'arxiu de entrada
if ((mapeo=mmap(0,stadbuf.st_size,PROT_READ,MAP_SHARED,fd,0))== MAP_FAILED)
error("Fallo al mapejar (open)");
close(fd); //cierro fitxer

//MOSTRAR MAPA
printf ("\n\n FITXER MAPEJAT EN MEMORIA\n");
system(path_maps); //Crida al sistema per a executar ordre

munmap(mapeo,stadbuf.st_size); //Elimine mapeig

printf ("\n\n ELIMINAT EL MAPEIG DEL FITXER EN MEMORIA\n");

system(path_maps);
exit(EXIT_SUCCESS);
}/**Fin de main**/
```

Es fa una crida per a executar l'ordre “cat /proc/%d/maps” creada i mostrar l'arxiu maps d'aquest procés

- Introducció
- Mapa de memòria en Linux
- Arxius mapejats en memòria
- **Biblioteques d'enllaç dinàmic**

- **Biblioteques o llibreries**

- Son arxius binaris que contenen codi de funcions o subprogrames
- Hi ha dues formes d'enllaçar-les al programa:
 - **Amb enllaç estàtic:** l'executable inclou tot el codi
 - Arxiu executable = Codi del programa + codi de funcions i llibreries
 - arxius amb extensió **.lib de Win32**, **.a de UNIX/LINUX**
 - **Enllaç dinàmic:** l'arxiu executable conté les referències a les rutines de biblioteca que utilitza i a l'àrea de memòria on s'emmagatzemaran
 - La seua càrrega en memòria es fa en invocar-les mitjançant el mapeig de les llibreries en memòria. Aquest treball el fa un programa carregador de les llibreries mitjançant la crida **dlopen**
 - Són els arxius **.dll de Win32** o **.lib de UNIX/LINUX**

- **Enllaç estàtic**

- Desavantatges
 - Executables grans
 - El codi de les funcions de la biblioteca està repetit en molts executables
 - Hi ha múltiples còpies del codi de les funcions de biblioteca en la memòria
 - Actualitzar la biblioteca implica tornar a montar
- Avantatges
 - L'executable conté tot el codi que cal per a la seua execució

- **Enllaç dinàmic**

- Avantatges
 - Executables més curts, estalvi d'espai en el disc i en la memòria
 - Codi de les rutines de biblioteca només en el fitxer de biblioteca
 - Els processos poden compartir codi de biblioteca
 - Actualització automàtica de les biblioteques: Ús de les versions
- Desavantatges
 - L'enllaç es resol en temps d'execució, i això pot ocasionar retards en l'execució

- Programa exemple1

- Executable generat amb **enllaç estàtic** de la llibreria *math*

```
pblanes$ gcc exemple1.c -static -o exemple1 -lm
```

```
pblanes$ ls -l
```

```
total 660
```

```
-rwxr-xr-x 1 pblanes disca-upvnet 670227 2011-10-20 15:54 exemple1
```

```
-rw-r--r-- 1 pblanes disca-upvnet 905 2011-10-20 13:30 exemple1.c
```

- Executable generat amb **enllaç dinàmic** de la llibreria *math*

```
pblanes$ gcc exemple1.c -o exemple1 -lm
```

```
pblanes$ ls -l
```

```
total 16
```

```
-rwxr-xr-x 1 pblanes disca-upvnet 10301 2011-10-20 15:56 exemple1
```

```
-rw-r--r-- 1 pblanes disca-upvnet 905 2011-10-20 13:30 exemple1.c
```