



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMAS OPERATIVOS

Grupo 6

PROYECTO NO. 3

Asignación de Memoria a un Sistema Real

ALUMNOS

Abad Vásquez, Aldo

Rosales López, Luis André

PROFESOR

Ing. Gunnar Eyal Wolf Iszaevich



Ciudad Universitaria, Cd. Mx., 8 de mayo de 2022

Proyecto 3: Asignación de Memoria a un Sistema Real

Introducción

La paginación consiste en la división de procesos en varios bloques de tamaño fijo (más pequeños que los segmentos) llamados páginas, dejando de requerir de esta forma que la asignación sea de un área contigua de memoria. En la paginación no basta sólo con indicar dónde inicia y termina el área de memoria de cada proceso, sino que se debe establecer un mapeo entre la ubicación real (física) y la presentada a cada uno de los procesos (lógica). La memoria se presentará a cada proceso como si fuera de su uso exclusivo.

La memoria física se divide en una serie de marcos (*frames*), todos ellos del mismo tamaño, y el espacio para cada proceso se divide en una serie de páginas (*pages*), del mismo tamaño que los marcos. La MMU se encarga del mapeo entre páginas y marcos mediante tablas de páginas.

Cuando se trabaja bajo una arquitectura que maneja paginación, las direcciones que maneja el CPU ya no son presentadas de forma absoluta. Los bits de cada dirección se separan en un identificador de página y un desplazamiento, de forma similar a lo presentado al hablar de resolución de instrucciones en tiempo de ejecución. La principal diferencia con lo entonces abordado es que cada proceso tendrá ya no un único espacio en memoria, sino una multitud de páginas.

Planteamiento del proyecto

Crear un programa ejecutable al cual se le indique un *PID*, *ID* del proceso que estemos analizando, y que genere con este un mapa.

El programa debe recibir como entrada el mapa de memoria de cualquier proceso, al que tenga permiso de lectura, y entregar:

- Identificación de las secciones en memoria, indicando el tamaño de cada una de ellas. Además de las regiones correspondientes a bibliotecas de sistema y a otros archivos mapeados a memoria.
- Representación visual de la memoria del proceso.

Implementación

Descripción del Entorno de Desarrollo

Lenguaje y versión utilizados. El lenguaje empleado para la implementación es Python 3 (3.7.3). Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Bibliotecas utilizadas

- **Lib/re.py** : este módulo proporciona operaciones de coincidencia de expresiones regulares similares a las que se encuentran en Perl.
- **itertools** : este módulo implementa una serie de bloques de construcción de iteradores inspirados en construcciones de APL, Haskell y SML. Cada uno de ellos ha sido refundido en una forma adecuada para Python.
- **Lib/argparse.py** : este módulo facilita la escritura de interfaces de línea de comandos fáciles de usar. El programa define los argumentos que necesita y argparse se encarga de analizarlos en sys.argv. El módulo argparse también genera automáticamente mensajes de ayuda y de uso y emite errores cuando los usuarios dan al programa argumentos no válidos.
- **colorama** : este módulo hace que las secuencias de caracteres de escape ANSI (para producir texto de terminal coloreado y posicionamiento del cursor) funcionen bajo MS Windows.
- **Lib/dataclasses.py** : este módulo proporciona un decorador y funciones para añadir automáticamente métodos especiales generados como `__init__()` y `__repr__()` a las clases definidas por el usuario. Fue descrito originalmente en PEP 557.

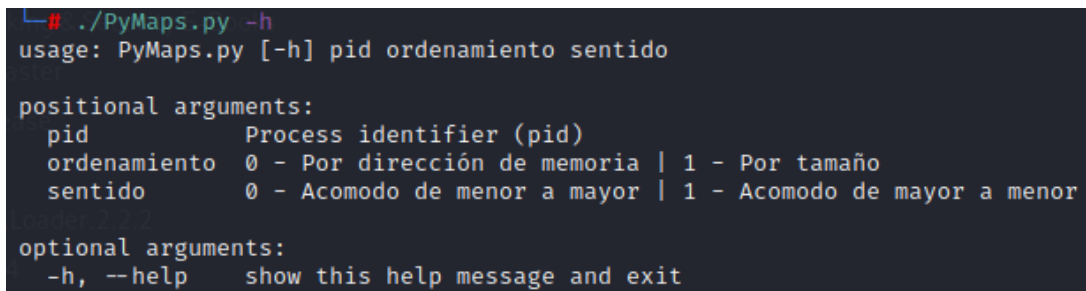
Las variables miembro a utilizar en estos métodos generados se definen utilizando anotaciones de tipo PEP 526.

Sistema operativo / distribución de desarrollo y/o pruebas. El desarrollo se realizó y probó en Debian 10, con ayuda del editor de texto Vim y con ejecución en consola. Igualmente, se probó su funcionamiento en Windows 11 mediante el *IDLE* de *Python*. Es importante que el usuario que desee utilizar el programa se asegure de tener instalado el intérprete de Python (Python3) así como las bibliotecas mencionadas anteriormente. Esto se puede hacer con el comando:

```
$ pip install <biblioteca>
```

Después de esto es posible ejecutar el programa con el siguiente comando:

```
$/PyMaps.py <PID> <Ordenamiento> <Sentido>
```



```
└─$ ./PyMaps.py -h
usage: PyMaps.py [-h] pid ordenamiento sentido

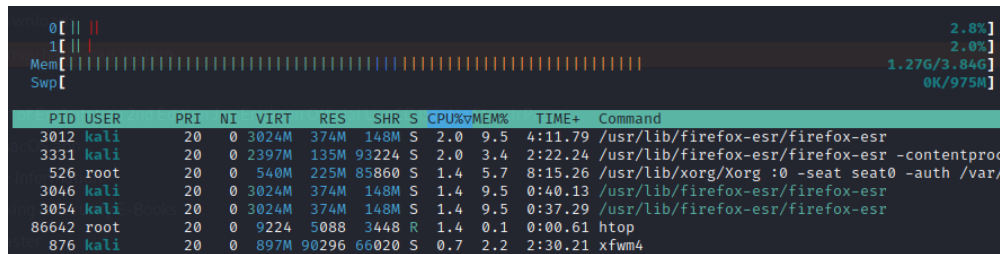
positional arguments:
  pid                Process identifier (pid)
  ordenamiento       0 - Por dirección de memoria | 1 - Por tamaño
  sentido            0 - Acomodo de menor a mayor | 1 - Acomodo de mayor a menor

optional arguments:
  -h, --help         show this help message and exit
```

El programa cuenta además con información de ayuda al ejecutarlo con la opción -h. En cuanto a los parámetros que recibe, es importante destacar que el programa permite ordenar los resultados según la dirección en memoria de cada bloque, así como por el tamaño de cada uno de ellos. Además, permite ordenar en un sentido ascendente o descendente, según desee el usuario.

Evidencias de Ejecución

Antes de proceder con la ejecución del programa es necesario identificar el PID del proceso que queramos analizar. Para ello nos ayudamos del comando `htop`. De entre los procesos mostrados a continuación nos centraremos en analizar a dos: El proceso con PID 3012 (`firefox-esr`) y el proceso con PID 876 (`xfwm4`).



PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
3012	kali	20	0	3024M	374M	148M	S	2.0	9.5	4:11.79	/usr/lib/firefox-esr/firefox-esr
3331	kali	20	0	2397M	135M	93224	S	2.0	3.4	2:22.24	/usr/lib/firefox-esr/firefox-esr -contentproc
526	root	20	0	540M	225M	85860	S	1.4	5.7	8:15.26	/usr/lib/xorg/Xorg :0 -seat seat0 -auth /var/
3046	kali	20	0	3024M	374M	148M	S	1.4	9.5	0:40.13	/usr/lib/firefox-esr/firefox-esr
3054	kali	20	0	3024M	374M	148M	S	1.4	9.5	0:37.29	/usr/lib/firefox-esr/firefox-esr
86642	root	20	0	9224	5088	3448	R	1.4	0.1	0:00.61	htop
876	kali	20	0	897M	90296	66020	S	0.7	2.2	2:30.21	xfwm4

Caso 1: firefox-esr (PID: 3012)

El primer caso resulta interesante por la cantidad de memoria virtual utilizada (3GB). Siguiendo los requerimientos planteados para el proyecto, nuestro programa identifica las siguientes características de cada bloque:

- Uso

El uso se refiere, como su nombre lo indica, al uso que se le está dando a determinado bloque. A cada uso le corresponde un color diferente:

- **Magenta:** Bloques usados por bibliotecas, ya sea como datos o texto.
- Verde: Bloque usado por el stack. Es esperable encontrarlo en posiciones altas de memoria.
- **Amarillo:** Bloque usado por el heap. Es curioso que para este primer caso no encontramos ningún bloque asignado para el heap.
- **Azul:** Bloques usados para datos en general, pudiendo ser archivos de caché, fuentes, entre otros. Otra forma de categorizar estos bloques es en segmentos de datos inicializados y segmentos de datos sin inicializar.
- **Rojo:** Bloques “vacíos” de uso anónimo.
- **Cyan:** Bloques de texto que contienen código ejecutable.

Para definir el tipo de uso que tiene cierto segmento de memoria tomamos como base los permisos que tiene asignado, así como el mapeo que le corresponde.

- *Dirección en memoria inicial (De pág)*

Dirección de inicio del mapeo. Se utiliza para ordenar la salida así como para determinar el tamaño del segmento.

- *Dirección en memoria final (A pág)*

Dirección de final del mapeo. Se utiliza para ordenar la salida así como para determinar el tamaño del segmento.

- *Tamaño*

Se calcula a partir de la diferencia entre la dirección en memoria final menos la dirección en memoria inicial.

- *Páginas*

Asumimos un tamaño de página de 4Kb, por lo que es posible obtener el número de páginas usadas dividiendo el tamaño entre 4Kb = 4096 Bytes. Sería posible determinar el tamaño de página usado por el kernel en el archivo /proc/PID/smmaps, así como el número de páginas utilizadas (RSS), así como el número de páginas compartidas (PSS), aunque estas métricas toman en cuenta otros parámetros para arrojar sus resultados.

- *Permisos*

Cada segmento está mapeado con un conjunto de permisos que controla que acciones se permite realizar sobre ellos. Estos permisos, también conocidos como modos, son los siguientes:

- **Read-only (r)** – Significa que el segmento es legible. De hecho, lo más usual es que todos los segmentos tengan este modo.
- **Read-write (w)** – Significa que el segmento es legible y escribible para permitir que los datos contenidos en él se modifiquen.
- **Execute (x)** – Significa que el segmento contiene código ejecutable.

Nosotros utilizamos los modos anteriores para determinar el uso de cada segmento, según la siguiente lógica:

- **Datos** si sus permisos son de lectura, o lectura y escritura más no de ejecución. (r—o rw-)
- **Texto** si sus permisos son de lectura y ejecución, más no de escritura (r-x)
- **Privados** si no tiene permisos de lectura, escritura o ejecución. Cabe aclarar que esto es una asunción, pues no toma en cuenta el cuarto permiso.

Es importante que existen dos modos adicionales a los mencionados anteriormente, los cuales son:

- **Private (p)** – Significa que el segmento es privado y solo es visible desde el proceso que lo apartó.
- **Shared (s)** – Significa que múltiple (al menos 2) procesos comparten dicho segmento.

Datos	7fe1b3cf7000	7fe1b3e7e000	1.5M	391 págs	r--	/home/kali/.cache/mozilla/firefox/3tlgq9qf.default-esr/startupCache/startupCache.8.little
Anónimo	7fe1b3cb7000	7fe1b3cf7000	256.0k	64 págs	rw-	— Vacío —
Anónimo	7fe1b3cb6000	7fe1b3cb7000	4.0k	1 págs	---	— Vacío —
Bib → datos	7fe1b3cb5000	7fe1b3cb6000	4.0k	1 págs	rw-	/usr/lib/x86_64-linux-gnu/gio/modules/libgvfsdbus.so
Bib → datos	7fe1b3cb3000	7fe1b3cb5000	8.0k	2 págs	r--	/usr/lib/x86_64-linux-gnu/gio/modules/libgvfsdbus.so
Bib → priv.	7fe1b3cb2000	7fe1b3cb3000	4.0k	1 págs	---	/usr/lib/x86_64-linux-gnu/gio/modules/libgvfsdbus.so
Bib → datos	7fe1b3ca7000	7fe1b3cb2000	44.0k	11 págs	r--	/usr/lib/x86_64-linux-gnu/gio/modules/libgvfsdbus.so
Bib → texto	7fe1b3c8000	7fe1b3ca7000	124.0k	31 págs	r-x	/usr/lib/x86_64-linux-gnu/gio/modules/libgvfsdbus.so
Bib → datos	7fe1b3c7d000	7fe1b3c8000	44.0k	11 págs	r--	/usr/lib/x86_64-linux-gnu/gio/modules/libgvfsdbus.so
Bib → datos	7fe1b3c7c000	7fe1b3c7d000	4.0k	1 págs	rw-	/usr/lib/x86_64-linux-gnu/gvfs/libgvfscommon.so
Bib → datos	7fe1b3c76000	7fe1b3c7c000	24.0k	6 págs	r--	/usr/lib/x86_64-linux-gnu/gvfs/libgvfscommon.so
Bib → priv.	7fe1b3c75000	7fe1b3c76000	4.0k	1 págs	---	/usr/lib/x86_64-linux-gnu/gvfs/libgvfscommon.so
Bib → datos	7fe1b3c68000	7fe1b3c75000	52.0k	13 págs	r--	/usr/lib/x86_64-linux-gnu/gvfs/libgvfscommon.so
Bib → texto	7fe1b3c4e000	7fe1b3c68000	104.0k	26 págs	r-x	/usr/lib/x86_64-linux-gnu/gvfs/libgvfscommon.so
Bib → datos	7fe1b3c3b000	7fe1b3c4e000	76.0k	19 págs	r--	/usr/lib/x86_64-linux-gnu/gvfs/libgvfscommon.so
Anónimo	7fe1b3c1b000	7fe1b3c3b000	128.0k	32 págs	rw-	— Vacío —
Anónimo	7fe1b3c1a000	7fe1b3c1b000	4.0k	1 págs	---	— Vacío —
Bib → datos	7fe1b3c19000	7fe1b3c1a000	4.0k	1 págs	rw-	/usr/lib/x86_64-linux-gnu/libasyns.so.0.3.1
Bib → datos	7fe1b3c18000	7fe1b3c19000	4.0k	1 págs	r--	/usr/lib/x86_64-linux-gnu/libasyns.so.0.3.1
Bib → datos	7fe1b3c17000	7fe1b3c18000	4.0k	1 págs	r--	/usr/lib/x86_64-linux-gnu/libasyns.so.0.3.1
Bib → texto	7fe1b3c14000	7fe1b3c17000	12.0k	3 págs	r-x	/usr/lib/x86_64-linux-gnu/libasyns.so.0.3.1
Bib → datos	7fe1b3c12000	7fe1b3c14000	8.0k	2 págs	r--	/usr/lib/x86_64-linux-gnu/libasyns.so.0.3.1
Bib → datos	7fe1b3c11000	7fe1b3c12000	4.0k	1 págs	rw-	/usr/lib/x86_64-linux-gnu/libwrap.so.0.7.6
Bib → datos	7fe1b3c10000	7fe1b3c11000	4.0k	1 págs	r--	/usr/lib/x86_64-linux-gnu/libwrap.so.0.7.6
Bib → datos	7fe1b3c0e000	7fe1b3c10000	8.0k	2 págs	r--	/usr/lib/x86_64-linux-gnu/libwrap.so.0.7.6
Bib → texto	7fe1b3c09000	7fe1b3c0e000	20.0k	5 págs	r-x	/usr/lib/x86_64-linux-gnu/libwrap.so.0.7.6
Bib → datos	7fe1b3c06000	7fe1b3c09000	12.0k	3 págs	r--	/usr/lib/x86_64-linux-gnu/libwrap.so.0.7.6
Datos	7fe1b3c01000	7fe1b3c06000	20.0k	5 págs	r--	/var/cache/fontconfig/eb7344a5-adb2-4f28-8ef6-ed800338d4c-le64.cache-7
Datos	7fe1b3c00000	7fe1b3c01000	4.0k	1 págs	r--	/memfd:mozilla-ipc (deleted)
Anónimo	7fe1b3bf1000	7fe1b3c00000	4.0k	1 págs	---	— Vacío —

• Uso o Mapeo

La ruta del archivo correspondiente al mapeo. También puede ocurrir que no se refiera a un archivo, sino a una sección de memoria de tipo:

○ **Heap**

Espacio en memoria solicitado explícitamente por el programador.

○ **Stack**

Almacena datos temporales producto de las llamadas a funciones, guarda las variables locales, parámetros de llamada a función, direcciones de retorno, entre otros.

- **Vdso**

Segmento de memoria correspondiente a una biblioteca exportada por el kernel para acelerar la ejecución de ciertas llamadas al sistema que no necesariamente requieren correr en el espacio del kernel.

- **Vacío**

Segmentos con contenido sin relevancia para el proceso que puede ser utilizado en un futuro.

```

-# ./PyMaps.py 3012 0 1
#### Reporte de memoria para el proceso PID = 3012 | Total de memoria utilizada: 3.0G ####

```

	Uso	De pág	A pág	Tamaño	Páginas	Permisos	Uso o Mapeo
	Vdso	7ffe927e6000	7ffe927e8000	8.0k	2 págs	r-x	[vdso]
	Vvar	7ffe927e2000	7ffe927e6000	16.0k	4 págs	r--	[vvar]
	Stack	7ffe927b7000	7ffe927da000	140.0k	35 págs	rw-	[stack]
Bib → datos	7felc4a48000	7felc4a4a000	8.0k	2 págs	rw-		/usr/lib/x86_64-linux-gnu/ld-2.33.so
Bib → datos	7felc4a46000	7felc4a48000	8.0k	2 págs	r--		/usr/lib/x86_64-linux-gnu/ld-2.33.so
Bib → datos	7felc4a3c000	7felc4a46000	40.0k	10 págs	r--		/usr/lib/x86_64-linux-gnu/ld-2.33.so
Bib → texto	7felc4a18000	7felc4a3c000	144.0k	36 págs	r-x		/usr/lib/x86_64-linux-gnu/ld-2.33.so
Bib → datos	7felc4a17000	7felc4a18000	4.0k	1 págs	r--		/usr/lib/x86_64-linux-gnu/ld-2.33.so
Bib → datos	7felc4a16000	7felc4a17000	4.0k	1 págs	rw-		/usr/lib/firefox-esr/liblpllibs.so
Bib → datos	7felc4a15000	7felc4a16000	4.0k	1 págs	r--		/usr/lib/firefox-esr/liblpllibs.so
Bib → datos	7felc4a13000	7felc4a15000	8.0k	2 págs	r--		/usr/lib/firefox-esr/liblpllibs.so
Bib → texto	7felc4a0e000	7felc4a13000	20.0k	5 págs	r-x		/usr/lib/firefox-esr/liblpllibs.so
Bib → datos	7felc4a0b000	7felc4a0e000	12.0k	3 págs	r--		/usr/lib/firefox-esr/liblpllibs.so
Bib → datos	7felc4a0a000	7felc4a0b000	4.0k	1 págs	rw-		/usr/lib/firefox-esr/libmozgtk.so
Bib → datos	7felc4a09000	7felc4a0a000	4.0k	1 págs	r--		/usr/lib/firefox-esr/libmozgtk.so
Bib → datos	7felc4a08000	7felc4a09000	4.0k	1 págs	r--		/usr/lib/firefox-esr/libmozgtk.so
Bib → texto	7felc4a07000	7felc4a08000	4.0k	1 págs	r-x		/usr/lib/firefox-esr/libmozgtk.so
Bib → datos	7felc4a06000	7felc4a07000	4.0k	1 págs	r--		/usr/lib/firefox-esr/libmozgtk.so
Datos	7felc4a05000	7felc4a06000	4.0k	1 págs	r--		/usr/share/locale/en/LC_MESSAGES/gtk30.mo
Anónimo	7felc49ff000	7felc4a05000	24.0k	6 págs	rw-		— Vacío —
Bib → datos	7felc49fe000	7felc49ff000	4.0k	1 págs	rw-		/usr/lib/x86_64-linux-gnu/libpthread-2.33.so
Bib → datos	7felc49fd000	7felc49fe000	4.0k	1 págs	r--		/usr/lib/x86_64-linux-gnu/libpthread-2.33.so
Bib → datos	7felc49f8000	7felc49fd000	20.0k	5 págs	r--		/usr/lib/x86_64-linux-gnu/libpthread-2.33.so
Bib → texto	7felc49e9000	7felc49f8000	60.0k	15 págs	r-x		/usr/lib/x86_64-linux-gnu/libpthread-2.33.so
Bib → datos	7felc49e2000	7felc49e9000	28.0k	7 págs	r--		/usr/lib/x86_64-linux-gnu/libpthread-2.33.so

Caso 2: xfwm4 (PID: 876)

En este segundo caso vemos desplegado el mapeo de la memoria con respecto a las direcciones de memoria en orden descendente. No es sorpresa encontrar el stack entre las primeras posiciones.

```

-# ./PyMaps.py 876 0 1
#### Reporte de memoria para el proceso PID = 876 | Total de memoria utilizada: 897.2M ####

```

	Uso	De pág	A pág	Tamaño	Páginas	Permisos	Uso o Mapeo
	Vdso	7ffc9bbfc000	7ffc9bbfe000	8.0k	2 págs	r-x	[vdso]
	Vvar	7ffc9bbfb000	7ffc9bbfc000	16.0k	4 págs	r--	[vvar]
	Stack	7ffc9bbbd000	7ffc9bbbd3000	140.0k	35 págs	rw-	[stack]
Bib → datos	7f83d2448000	7f83d244a000	8.0k	2 págs	rw-		/usr/lib/x86_64-linux-gnu/ld-2.33.so
Bib → datos	7f83d2446000	7f83d2448000	8.0k	2 págs	r--		/usr/lib/x86_64-linux-gnu/ld-2.33.so
Bib → datos	7f83d243c000	7f83d2446000	40.0k	10 págs	r--		/usr/lib/x86_64-linux-gnu/ld-2.33.so
Bib → texto	7f83d2418000	7f83d243c000	144.0k	36 págs	r-x		/usr/lib/x86_64-linux-gnu/ld-2.33.so
Bib → datos	7f83d2417000	7f83d2418000	4.0k	1 págs	r--		/usr/lib/x86_64-linux-gnu/ld-2.33.so
Datos	7f83d2416000	7f83d2417000	4.0k	1 págs	r--		/usr/share/locale/en/LC_MESSAGES/gtk30.mo
Datos	7f83d2415000	7f83d2416000	4.0k	1 págs	r--		/usr/share/locale/en/LC_MESSAGES/gtk30-properties.mo
Datos	7f83d2412000	7f83d2415000	12.0k	3 págs	r--		/var/cache/fontconfig/85c66712-dd64-4f97-bcce-cbce0e933ca8-le64.cache-7
Datos	7f83d240f000	7f83d2412000	12.0k	3 págs	r--		/var/cache/fontconfig/491973fa-06c0-41f2-bba2-4f7e151cd7b1-le64.cache-7
Datos	7f83d240e000	7f83d240f000	4.0k	1 págs	r--		/var/cache/fontconfig/8e61bf6e-833e-4542-9ede-9212d2a2b460-le64.cache-7
Datos	7f83d240d000	7f83d240e000	4.0k	1 págs	r--		/var/cache/fontconfig/75027219-14bb-475e-8743-7d141a32de8b-le64.cache-7
Datos	7f83d240a000	7f83d240d000	12.0k	3 págs	r--		/var/cache/fontconfig/26d6119b-f9f8-4bea-8469-b04b52e23f1d-le64.cache-7
Datos	7f83d2409000	7f83d240a000	4.0k	1 págs	r--		/var/cache/fontconfig/13a843dc-a47f-4e91-a944-00622b8e07b7-le64.cache-7
Datos	7f83d2407000	7f83d2409000	8.0k	2 págs	r--		/var/cache/fontconfig/14e86ced-05f3-401d-9021-8388bb694dd9-le64.cache-7
Datos	7f83d2406000	7f83d2407000	4.0k	1 págs	r--		/var/cache/fontconfig/d9481301-2c77-4ba4f-9000-3ecf714d8de6-le64.cache-7
Anónimo	7f83d2405000	7f83d2406000	4.0k	1 págs	rw-		— Vacío —
Anónimo	7f83d2403000	7f83d2405000	8.0k	2 págs	rw-		— Vacío —

Una cosa interesante que encontramos al analizar el mapeo es la existencia de un segmento que está marcado como *deleted*. Nuestro programa lo ha marcado como una sección de datos, puesto que tiene permisos de escritura y lectura. Como vimos en clase, es probable que este archivo no esté eliminado de la memoria, sino más bien desligado del proceso (unlinked).

Datos	7f83ce814000	7f83ce81f000	44.0k	11 págs	r--	/usr/share/icons/hicolor/icon-theme.cache
Datos	7f83ce809000	7f83ce814000	44.0k	11 págs	r--	/usr/share/icons/hicolor/icon-theme.cache
Datos	7f83ce7ec000	7f83ce809000	116.0k	29 págs	r--	/usr/share/icons/Adwaita/icon-theme.cache
Datos	7f83ce7d1000	7f83ce7ec000	108.0k	27 págs	r--	/usr/share/fonts/opentype/cantarell/Cantarell-Bold.otf
Anónimo	7f83cdfd1000	7f83ce7d1000	8.0M	2048 págs	rw-	— Vacío —
Anónimo	7f83cdfd0000	7f83cdfd1000	4.0k	1 págs	---	— Vacío —
Anónimo	7f83cd7d0000	7f83cdfd0000	8.0M	2048 págs	rw-	— Vacío —
Anónimo	7f83cd7cf000	7f83cd7d0000	4.0k	1 págs	---	— Vacío —
Datos	7f83cd74f000	7f83cd7cf000	512.0k	128 págs	rw-	/SYSV00000000 (deleted)
Anónimo	7f83cd74e000	7f83cd74f000	4.0k	1 págs	rw-	— Vacío —
Bib → datos	7f83cd74d000	7f83cd74e000	4.0k	1 págs	rw-	/usr/lib/x86_64-linux-gnu/libGL.so.1.7.0
Bib → datos	7f83cd73f000	7f83cd74d000	56.0k	14 págs	r--	/usr/lib/x86_64-linux-gnu/libGL.so.1.7.0
Bib → datos	7f83cd72a000	7f83cd73f000	84.0k	21 págs	r--	/usr/lib/x86_64-linux-gnu/libGL.so.1.7.0
Bib → texto	7f83cd70b000	7f83cd72a000	124.0k	31 págs	r-x	/usr/lib/x86_64-linux-gnu/libGL.so.1.7.0
Bib → datos	7f83cd6c8000	7f83cd70b000	268.0k	67 págs	r--	/usr/lib/x86_64-linux-gnu/libGL.so.1.7.0
Anónimo	7f83cd6c0000	7f83cd6c8000	32.0k	8 págs	rw-	— Vacío —
Bib → datos	7f83cd6bf000	7f83cd6c0000	4.0k	1 págs	rw-	/usr/lib/x86_64-linux-gnu/libGldispatch.so.0.0.0
Bib → datos	7f83cd6a3000	7f83cd6bf000	112.0k	28 págs	r--	/usr/lib/x86_64-linux-gnu/libGldispatch.so.0.0.0
Bib → datos	7f83cd68f000	7f83cd6a3000	80.0k	20 págs	r--	/usr/lib/x86_64-linux-gnu/libGldispatch.so.0.0.0
Bib → texto	7f83cd650000	7f83cd68f000	252.0k	63 págs	r-x	/usr/lib/x86_64-linux-gnu/libGldispatch.so.0.0.0
Bib → datos	7f83cd610000	7f83cd650000	256.0k	64 págs	r--	/usr/lib/x86_64-linux-gnu/libGldispatch.so.0.0.0
Anónimo	7f83cd60f000	7f83cd610000	4.0k	1 págs	rw-	— Vacío —

También resulta relevante mencionar que en el mapeo de este proceso sí encontramos una sección dedicada al heap. Esto quiere decir que hay una petición explícita de memoria dentro del programa que se está ejecutando en dicho proceso.

Bib → datos	7f83a3ffe000	7f83a4000000	8.0k	2 págs	rw-	/usr/lib/x86_64-linux-gnu/librsvg-2.0.2.48.0
Bib → datos	7f83a3f32000	7f83a3ffe000	816.0k	204 págs	r--	/usr/lib/x86_64-linux-gnu/librsvg-2.0.2.48.0
Bib → datos	7f83a3c3a000	7f83a3f32000	3.0M	760 págs	r--	/usr/lib/x86_64-linux-gnu/librsvg-2.0.2.48.0
Bib → texto	7f83a3668000	7f83a3c3a000	5.8M	1490 págs	r-x	/usr/lib/x86_64-linux-gnu/librsvg-2.0.2.48.0
Bib → datos	7f83a354f000	7f83a3668000	1.1M	281 págs	r--	/usr/lib/x86_64-linux-gnu/librsvg-2.0.2.48.0
Anónimo	7f83a2d4f000	7f83a354f000	8.0M	2048 págs	rw-	— Vacío —
Anónimo	7f83a2d4e000	7f83a2d4f000	4.0k	1 págs	---	— Vacío —
Anónimo	7f83a254e000	7f83a2d4e000	8.0M	2048 págs	rw-	— Vacío —
Anónimo	7f83a254d000	7f83a254e000	4.0k	1 págs	---	— Vacío —
Anónimo	7f839c021000	7f83a0000000	63.9M	16351 págs	---	— Vacío —
Anónimo	7f839c000000	7f839c021000	132.0k	33 págs	rw-	— Vacío —
Anónimo	7f8394021000	7f839c000000	63.9M	16351 págs	---	— Vacío —
Anónimo	7f8394000000	7f8394021000	132.0k	33 págs	rw-	— Vacío —
Heap	561fd229a000	561fd2c58000	9.7M	2494 págs	rw-	[heap]
Datos	561fd0b49000	561fd0b40000	8.0k	2 págs	rw-	/usr/bin/xfwm4
Datos	561fd0b47000	561fd0b49000	8.0k	2 págs	r--	/usr/bin/xfwm4
Datos	561fd0b31000	561fd0b47000	84.0k	21 págs	r--	/usr/bin/xfwm4
Texto	561fd0af3000	561fd0b31000	248.0k	62 págs	r-x	/usr/bin/xfwm4
Datos	561fd0ae5000	561fd0af3000	56.0k	14 págs	r--	/usr/bin/xfwm4

Finalmente, se muestra la misma información pero ahora ordenada de forma descendente según el tamaño de cada segmento.

### Reporte de memoria para el proceso PID = 876 Total de memoria utilizada: 897.2M ###							
	Uso	De pág	A pág	Tamaño	Páginas	Permisos	Uso o Mapeo
Bib	→ texto	7f83b235b000	7f83b78f3000	85.6M	21912 págs	r-x	/usr/lib/x86_64-linux-gnu/libLLVM-12.so.1
Anónimo		7f8394021000	7f8398000000	63.9M	16351 págs	---	— Vacío —
Anónimo		7f839c021000	7f83a0000000	63.9M	16351 págs	---	— Vacío —
Anónimo		7f83a4021000	7f83a8000000	63.9M	16351 págs	---	— Vacío —
Anónimo		7f83a8021000	7f83ac000000	63.9M	16351 págs	---	— Vacío —
Anónimo		7f83ac021000	7f83b0000000	63.9M	16351 págs	---	— Vacío —
Anónimo		7f83b8021000	7f83bc000000	63.9M	16351 págs	---	— Vacío —
Anónimo		7f83c8021000	7f83cc000000	63.9M	16351 págs	---	— Vacío —
Anónimo		7f83c0039000	7f83c4000000	63.8M	16327 págs	---	— Vacío —
Anónimo		7f83c404e000	7f83c8000000	63.7M	16306 págs	---	— Vacío —
Bib	→ datos	7f83b0844000	7f83b2359000	27.1M	6933 págs	r--	/usr/lib/x86_64-linux-gnu/libcudata.so.67.1
Bib	→ texto	7f83bd102000	7f83be2e8000	17.9M	4582 págs	r-x	/usr/lib/x86_64-linux-gnu/libz3.so.4
Bib	→ texto	7f83be80f000	7f83bf6cb000	14.7M	3772 págs	r-x	/usr/lib/x86_64-linux-gnu/dri/swrast_dri.so
Heap		561fd229a000	561fd2c58000	9.7M	2494 págs	rw-	[heap]
Anónimo		7f83a254e000	7f83a2d4e000	8.0M	2048 págs	rw-	— Vacío —
Anónimo		7f83a2d4f000	7f83a354f000	8.0M	2048 págs	rw-	— Vacío —
Anónimo		7f83bc077000	7f83bc877000	8.0M	2048 págs	rw-	— Vacío —
Anónimo		7f83bc878000	7f83bd078000	8.0M	2048 págs	rw-	— Vacío —

Referencias

- Wolf, G., Ruiz, E., Bergero, F., & Meza, E. (2015). *Fundamentos de Sistemas Operativos*. Universidad Nacional Autónoma de México, Instituto de Investigaciones Económicas : Facultad de Ingeniería. http://sistop.org/pdf/sistemas_operativos.pdf
- Miranda, M. (15 de mayo de 2020). *THE /proc FILESYSTEM* [Archivo TXT]. <https://mjmwired.net/kernel/Documentation/filesystems/proc.txt>
- Hoogland, F. (2017). *Linux Memory Explained* [Archivo PDF]. Accenture Operations. <https://fritshoogland.files.wordpress.com/2018/02/linux-memory-explained.pdf>
- Baeldung. (8 de abril de 2022). *Understanding the Linux /proc/id/maps File*. <https://www.baeldung.com/linux/proc-id-maps>
- *Shared_dirty vs Private_dirty in shared memory*. (2015, 8 octubre). Stack Overflow. Recuperado 5 de mayo de 2022, de <https://stackoverflow.com/questions/33027341/shared-dirty-vs-private-dirty-in-shared-memory/33029606#33029606>
- Hunt, J. (s. f.). *maps, smaps and Memory Stats!* James Hunt (.Us). Recuperado 6 de mayo de 2022, de <https://jameshunt.us/writings/smaps/>
- *Memory - Part 2: Understanding Process memory* . (s. f.). Intersec TechTalk. Recuperado 2 de mayo de 2022, de <https://techtalk.intersec.com/2013/07/memory-part-2-understanding-process-memory/>