

Práctica 5 de Sistemas Operativos: Mapa de memoria de un proceso

Autores: Sergio Núñez González y Santiago Vilas Pampín

EJERCICIO 1:

```
PID: 612278
Dirección de la función main: 0x582b8b6e01c9
Matriz p creada
Dirección base del array p: 0x7fffd8fc4f80
Pausa para poder mirar el mapa (pulsa enter para seguir)

p[0][0][0]: 0x7fffd8fc4f80
p[0][0][1]: 0x7fffd8fc4f84
p[0][0][9]: 0x7fffd8fc4fa4
p[0][1][0]: 0x7fffd8fc4fa8
p[1][0][0]: 0x7fffd8fc5110
p[9][9][9]: 0x7fffd8fc5f1c
local_c: 0x7fffd8fc4f77
local_i: 0x7fffd8fc4f78
local_f: 0x7fffd8fc4f7c
Pausa para poder mirar el mapa (pulsa enter para seguir)

Variables Globales (DATA/BSS)
c: 0x582b8b6e3010
i: 0x582b8b6e3014
pi: 0x582b8b6e3018
arr: 0x582b8b6e3040
arr[último]: 0x582b8b7a653c
unused_global: 0x582b8b7a6540
Pausa para poder mirar el mapa (pulsa enter para seguir)
```

En este apartado se ha creado un programa para analizar la ubicación en memoria de diferentes tipos de variables globales y locales, inicializadas y no inicializadas, además de la función principal main. Se ha examinado el mapa de memoria del proceso para identificar las regiones de memoria correspondientes. La función main se encuentra en la dirección 0x582b8b6e01c9, correspondiente al segmento de código ejecutable (582b8b6e0000-582b8b6e1000 r-xp).

Las variables globales inicializadas (c, i, pi) aparecen en la región de Datos (rw-p) respaldada por el archivo, desde la dirección

0x582b8b6e3010-0x582b8b6e3018. Luego tiene lugar un salto para las variables globales no inicializadas, las cuales empiezan a partir de la dirección 0x582b8b6e3040 en vez de en la dirección de memoria inmediatamente contigua: 0x582b8b6e301c.

```
582b8b6df000-582b8b6e0000 r--p 00000000 103:06 306013 /home/san/main
582b8b6e0000-582b8b6e1000 r-xp 00001000 103:06 306013 /home/san/main
582b8b6e1000-582b8b6e2000 r--p 00002000 103:06 306013 /home/san/main
582b8b6e2000-582b8b6e3000 r--p 00002000 103:06 306013 /home/san/main
582b8b6e3000-582b8b6e4000 rw-p 00003000 103:06 306013 /home/san/main
582b8b6e4000-582b8b7a7000 rw-p 00000000 00:00 0
582ba73b7000-582ba73d8000 rw-p 00000000 00:00 0 [heap]
7a7b76400000-7a7b76428000 r--p 00000000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
7a7b76428000-7a7b765b0000 r-xp 00028000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
7a7b765b0000-7a7b765ff000 r--p 001b0000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
7a7b765ff000-7a7b76603000 r--p 001fe000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
7a7b76603000-7a7b76605000 rw-p 00202000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
7a7b76605000-7a7b76612000 rw-p 00000000 00:00 0
7a7b76693000-7a7b76696000 rw-p 00000000 00:00 0
7a7b766ae000-7a7b766b0000 rw-p 00000000 00:00 0
7a7b766b0000-7a7b766b2000 r--p 00000000 00:00 0 [vvar]
7a7b766b2000-7a7b766b4000 r--p 00000000 00:00 0 [vvar_vclock]
7a7b766b4000-7a7b766b6000 r-xp 00000000 00:00 0 [vdso]
7a7b766b6000-7a7b766b7000 r--p 00000000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7a7b766b7000-7a7b766e2000 r-xp 00001000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7a7b766e2000-7a7b766ec000 r--p 0002c000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7a7b766ec000-7a7b766ee000 r--p 00036000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7a7b766ee000-7a7b766ff000 rw-p 00038000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7fffd8fa7000-7fffd8fc8000 rw-p 00000000 00:00 0 [stack]
fffffffff600000-fffffffff601000 --xp 00000000 00:00 0 [vsyscall]
```

En el extremo opuesto del espacio de direcciones, las variables locales se encuentran en direcciones muy altas (la primera posición de p) empezando en la dirección más alta del [stack], 0x0x7fffd8fc4f80, y crece hacia direcciones menores. Respecto al orden de almacenamiento del array tridimensional p, las direcciones obtenidas mostraron un incremento constante de 4 bytes (tamaño de int) entre elementos consecutivos (ej. p[0][0][0] a p[0][0][1]) y contigüidad entre el final de una fila y el inicio de la siguiente. Esto demuestra que los arrays multidimensionales se ordenan por filas, donde el índice más a la derecha es el que varía más rápidamente. Finalmente, se observó que el mapa de memoria permaneció estático durante toda la ejecución.

EJERCICIO 2:

```
PID: 239439
direccion de memoria de main: 0x64228145336d
direccion de memoria de f1: 0x6422814531e9
dirección de memoria de a en f1: 0x7ffc5122e95c
dirección de memoria de c (local) en f1: 0x7ffc5122e964
Valor de a en f1: 1
Valor de c en f1: 3
direccion de memoria de f2: 0x6422814532ab
dirección de memoria de b en f2: 0x7ffc5122e95c
dirección de memoria de d (local) en f2: 0x7ffc5122e964
Valor de b en f2: 2
Valor de d en f2: 4
dirección de memoria de a en main: 0x7ffc5122e980
dirección de memoria de b en main: 0x7ffc5122e984
Pausa para poder mirar el mapa (pulsa enter para seguir)
```

En este apartado se ha analizado el comportamiento de la memoria al invocar funciones y gestionar variables locales y parámetros. Se obtuvieron las direcciones de memoria de las funciones main, f1 y f2, así como de sus respectivas variables. Al contrastar estos datos con el archivo de mapa de memoria, se confirma que el código de las tres funciones

```
642281452000-642281453000 r--p 00000000 103:06 306050 /home/san/main2
642281453000-642281454000 r-xp 00001000 103:06 306050 /home/san/main2
642281454000-642281455000 r--p 00002000 103:06 306050 /home/san/main2
642281455000-642281456000 r--p 00002000 103:06 306050 /home/san/main2
642281456000-642281457000 rw-p 00003000 103:06 306050 /home/san/main2
6422a1cb6000-6422a1cd7000 rw-p 00000000 00:00 0 [heap]
70cb14a00000-70cb14a28000 r--p 00000000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
70cb14a28000-70cb14abb0000 r-xp 00028000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
70cb14bb0000-70cb14bff000 r--p 001b0000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
70cb14bff000-70cb14c03000 r--p 001fe000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
70cb14c03000-70cb14c05000 rw-p 00202000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
70cb14c05000-70cb14c12000 rw-p 00000000 00:00 0
70cb14c12000-70cb14c15000 rw-p 00000000 00:00 0
70cb14c15000-70cb14c16000 rw-p 00000000 00:00 0
70cb14c16000-70cb14c17000 r--p 00000000 00:00 0 [vvar]
70cb14c17000-70cb14c18000 r--p 00000000 00:00 0 [vvar_vclock]
70cb14c18000-70cb14c19000 r--p 00000000 00:00 0 [vdso]
70cb14c19000-70cb14c1a000 r--p 00000000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
70cb14c1a000-70cb14c1b000 r-xp 00001000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
70cb14c1b000-70cb14c1c000 r--p 0002c000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
70cb14c1c000-70cb14c1d000 r--p 00036000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
70cb14c1d000-70cb14caf000 rw-p 00038000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7ffc5120f000-7ffc51230000 rw-p 00000000 00:00 0 [stack]
fffffffff600000-fffffffff601000 --xp 00000000 00:00 0 [vsyscall]
```

reside en el segmento de código ejecutable, concretamente en la región con permisos de lectura y ejecución (r-xp) respaldada por el binario main2, ubicada en el rango de direcciones 0x642281453000.

Por otro lado, el análisis de las variables locales y los parámetros de las funciones reveló que todos ellos se almacenan en el segmento de la pila [stack] con direcciones altas (0x7ffc512...). Además, se observó que las variables locales de la función main se encuentran en direcciones superiores a las de las variables internas de f1 y f2, lo que es coherente con el comportamiento de la pila.

Se halló la coincidencia exacta de las direcciones de memoria de los parámetros y variables locales de las funciones f1 y f2. El parámetro a de f1 y el parámetro b de f2 ocuparon la dirección 0x7ffc5122e95c, mientras que las variables locales c y d compartieron la dirección 0x7ffc5122e964. Esto demuestra la reutilización de la pila: dado que las llamadas a las funciones se realizaron de manera secuencial en el main, los datos de f1 fueron liberados al finalizar su ejecución, permitiendo la reutilización del mismo espacio para los datos de f2.

EJERCICIO 3:

```
PID: 841085

Direccion de memoria del bloque reservado con malloc (pequeño): 0x622f2ccf7ac0
Última direccion de memoria del bloque reservado con malloc: 0x622f2ccf7c4c

Direccion de memoria del bloque reservado con malloc (mediano): 0x622f2ccf7c60
Última direccion de memoria del bloque reservado con malloc: 0x622f2ccf8bfc

Direccion de memoria del bloque reservado con alloca: 0x7ffc427dfe0
Última Direccion de memoria del bloque reservado con alloca: 0x7ffc427ee85c

Direccion de memoria del bloque reservado con malloc (antes realloc): 0x622f2ccf7c60
Última direccion de memoria del bloque reservado con malloc: 0x622f2cd0b4dc

Direccion de memoria del bloque reservado con realloc: 0x72e3f75d8010
Última direccion de memoria del bloque reservado con realloc: 0x72e3f75ff10c

Tamaño del puntero de malloc para double: 8 bytes
Direccion de memoria del bloque reservado con malloc: 0x622f2ccf7c60
Última Direccion de memoria del bloque reservado con malloc: 0x622f2ccf8bfc

Intentando liberar una dirección incorrecta (b + 1)...
free(): invalid pointer
[1] 841085 IOT instruction (core dumped) ./main3
```

En este apartado se ha examinado la gestión dinámica de memoria utilizando las llamadas al sistema *malloc*, *free*, *realloc* y la *alloca*. En la ejecución del proceso (PID 841085), se observó una clara distinción entre las regiones de memoria. Las reservas realizadas usando la función *malloc* devolvieron direcciones: 0x622f2c..., ubicadas en el [heap], pero la reserva que se realizó usando *alloca* devolvió la dirección: 0x7ffc427dfe0, que se encuentra en el [stack]. Se detectó que se hizo una eficaz

```
622f07e5b000-622f07e5c000 rw-p 00003000 103:06 306202 /home/san/main3
622f2ccf7000-622f2cd18000 rw-p 00000000 00:00 0 [heap]
72e3f7600000-72e3f7628000 r--p 00000000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
72e3f7628000-72e3f77b0000 r-xp 00028000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
72e3f77b0000-72e3f77ff000 r--p 001b0000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
72e3f77ff000-72e3f7803000 r--p 001fe000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
72e3f7803000-72e3f7805000 rw-p 00202000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
72e3f7805000-72e3f7812000 rw-p 00000000 00:00 0
72e3f7823000-72e3f7826000 rw-p 00000000 00:00 0
72e3f783d000-72e3f7840000 rw-p 00000000 00:00 0
72e3f7840000-72e3f7842000 r--p 00000000 00:00 0 [vvar]
```

```
622f07e57000-622f07e58000 r--p 00000000 103:06 306202 /home/san/main3
622f07e58000-622f07e59000 r-xp 00001000 103:06 306202 /home/san/main3
622f07e59000-622f07e5a000 r--p 00002000 103:06 306202 /home/san/main3
622f07e5a000-622f07e5b000 r--p 00002000 103:06 306202 /home/san/main3
622f07e5b000-622f07e5c000 rw-p 00003000 103:06 306202 /home/san/main3
622f2ccf7000-622f2cd18000 rw-p 00000000 00:00 0 [heap]
72e3f7600000-72e3f7628000 r--p 00000000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
72e3f7628000-72e3f77b0000 r-xp 00028000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
72e3f77b0000-72e3f77ff000 r--p 001b0000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
72e3f77ff000-72e3f7803000 r--p 001fe000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
72e3f7803000-72e3f7805000 rw-p 00202000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
72e3f7805000-72e3f7812000 rw-p 00000000 00:00 0
72e3f7823000-72e3f7826000 rw-p 00000000 00:00 0
72e3f783e000-72e3f7840000 rw-p 00000000 00:00 0
72e3f7840000-72e3f7842000 r--p 00000000 00:00 0 [vvar]
72e3f7842000-72e3f7844000 r--p 00000000 00:00 0 [vvar_vclock]
72e3f7844000-72e3f7846000 r-xp 00000000 00:00 0 [vdso]
72e3f7846000-72e3f7847000 r--p 00000000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
72e3f7847000-72e3f7872000 r-xp 00001000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
72e3f7872000-72e3f787c000 r--p 0002c000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
72e3f787c000-72e3f787e000 r--p 00036000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
72e3f787e000-72e3f7880000 rw-p 00038000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7ffc427cf000-7ffc427f0000 rw-p 00000000 00:00 0 [stack]
fffffffff600000-fffffffff601000 --xp 00000000 00:00 0 [vsyscall]
```

```
622f07e5b000-622f07e5c000 rw-p 00003000 103:06 306202 /home/san/main3
622f2ccf7000-622f2cd18000 rw-p 00000000 00:00 0 [heap]
72e3f75d8000-72e3f7600000 rw-p 00000000 00:00 0
72e3f7600000-72e3f7628000 r--p 00000000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
72e3f7628000-72e3f77b0000 r-xp 00028000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
72e3f77b0000-72e3f77ff000 r--p 001b0000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
72e3f77ff000-72e3f7803000 r--p 001fe000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
72e3f7803000-72e3f7805000 rw-p 00202000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
72e3f7805000-72e3f7812000 rw-p 00000000 00:00 0
72e3f7823000-72e3f7826000 rw-p 00000000 00:00 0
72e3f783e000-72e3f7840000 rw-p 00000000 00:00 0
```

reutilización de direcciones de memoria: tras liberar un bloque de tamaño medio (1000 ints) ubicado en 0x622f2ccf7c60, la siguiente solicitud de reserva (20000 ints) recibió exactamente la misma dirección de inicio, esto evita fragmentación innecesaria.

En el *realloc* al solicitar redimensionar el bloque a 40.000 ints (160 KB), la dirección de memoria cambió drásticamente de la región del [heap] a 0x72e3f75d8010. Coincidiendo con este cambio, la comparativa entre los mapas mostró la aparición de una nueva región anónima en el rango 72e3f75d8000-72e3f7600000. El tamaño de esta nueva región es de 163.840 bytes, lo que corresponde casi exactamente a los 160.000 bytes solicitados más la sobrecarga de gestión. Esto demuestra que, para solicitudes de memoria grandes que exceden ciertos umbrales, el sistema operativo opta por crear una nueva zona, fuera de la región del [heap].

Además, al aplicar el operador *sizeof* sobre el puntero devuelto por *malloc*, se obtuvo un valor de 8 bytes. Esto confirma que *sizeof* devuelve el tamaño del puntero (en arquitectura de 64 bits) y no el tamaño del bloque de memoria reservado.

Finalmente, como se indicaba en el enunciado, al intentar liberar un puntero modificado aritméticamente (puntero + 1), el sistema operativo detectó una inconsistencia en el bloque y abortó la ejecución con el error *free(): invalid pointer*, corroborando que la función *free* requiere estrictamente la dirección original devuelta por la reserva.

EJERCICIO 4:

```
Proceso padre (PID: 167815) antes del fork
(PADRE) Esperando a que el hijo termine...
Proceso hijo (PID: 168524) antes del malloc
malloc:0x633c34341ac0
Programa hijo ejecutando
Mi PID sigue siendo: 168524 (mismo que antes)
Pulsa ENTER para terminar el hijo...
```

En este apartado se estudió la evolución del mapa de memoria durante la creación de procesos y la sustitución de imagen mediante *exec*.

Al principio, tras la ejecución de la llamada al sistema *fork*, se comprobó que el mapa de memoria del proceso hijo (PID 168524) era idéntico al del proceso padre (PID 167815). Ambos procesos

```
633c2479e000-633c2479f000 r--p 00000000 103:06 306164 /home/san/main4
633c2479f000-633c247a0000 r-xp 00001000 103:06 306164 /home/san/main4
633c247a0000-633c247a1000 r--p 00002000 103:06 306164 /home/san/main4
633c247a1000-633c247a2000 r--p 00002000 103:06 306164 /home/san/main4
633c247a2000-633c247a3000 rw-p 00003000 103:06 306164 /home/san/main4
633c34341000-633c34362000 rw-p 00000000 00:00 0 [heap]
74d10ec00000-74d10ec28000 r--p 00000000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
74d10ec28000-74d10edb0000 r-xp 00028000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
74d10edb0000-74d10edff000 r--p 001b0000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
74d10edff000-74d10ee03000 r--p 001fe000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
74d10ee03000-74d10ee05000 rw-p 00202000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
74d10ee05000-74d10ee12000 rw-p 00000000 00:00 0
74d10ef67000-74d10ef6a000 rw-p 00000000 00:00 0
74d10ef82000-74d10ef84000 rw-p 00000000 00:00 0
74d10ef84000-74d10ef86000 r--p 00000000 00:00 0
74d10ef86000-74d10ef88000 r--p 00000000 00:00 0
74d10ef88000-74d10ef8a000 r-xp 00000000 00:00 0
74d10ef8a000-74d10ef8b000 r--p 00000000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
74d10ef8b000-74d10efb6000 r-xp 00001000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
74d10efb6000-74d10efc0000 r--p 0002c000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
74d10efc0000-74d10efc2000 r--p 00026000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
```


compartían exactamente las mismas regiones de memoria virtual, incluyendo el segmento de código ubicado en 0x633c247... y las librerías. Esto confirma que *fork* duplica el espacio de direcciones del padre (la “cajita”), creando una copia exacta. Incluso tras realizar una reserva de memoria con *malloc* en el hijo, el mapa de memoria se mantuvo estable dentro del mismo rango de direcciones del [heap] ya existente, sin requerir la creación de nuevas regiones visibles en el mapa.

```
633c2479e000-633c2479f000 r--p 00000000 103:06 306164 /home/san/main4
633c2479f000-633c247a0000 r-xp 00001000 103:06 306164 /home/san/main4
633c247a0000-633c247a1000 r--p 00002000 103:06 306164 /home/san/main4
633c247a1000-633c247a2000 r--p 00002000 103:06 306164 /home/san/main4
633c247a2000-633c247a3000 rw-p 00003000 103:06 306164 /home/san/main4
633c34341000-633c34362000 rw-p 00000000 00:00 0 [heap]
74d10ec00000-74d10ec28000 r--p 00000000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
74d10ec28000-74d10edb0000 r-xp 00028000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
74d10edb0000-74d10edff000 r--p 001b0000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
74d10edff000-74d10ee03000 r--p 001fe000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
74d10ee03000-74d10ee05000 rw-p 00202000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
74d10ee05000-74d10ee12000 rw-p 00000000 00:00 0
74d10ef67000-74d10ef6a000 rw-p 00000000 00:00 0
74d10ef82000-74d10ef84000 rw-p 00000000 00:00 0
74d10ef84000-74d10ef86000 r--p 00000000 00:00 0 [vvar]
74d10ef86000-74d10ef88000 r--p 00000000 00:00 0 [vvar_vclock]
74d10ef88000-74d10ef8a000 r-xp 00000000 00:00 0 [vdso]
74d10ef8a000-74d10ef8b000 r--p 00000000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
74d10ef8b000-74d10efb6000 r-xp 00001000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
74d10efb6000-74d10efc0000 r--p 0002c000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
74d10efc0000-74d10efc2000 r--p 00036000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
74d10efc2000-74d10efc4000 rw-p 00038000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7ffc00691000-7ffc006b2000 rw-p 00000000 00:00 0 [stack]
fffffffff600000-fffffffff601000 --xp 00000000 00:00 0 [vsyscall]
```

Tras la ejecución del *execv*, se observó un cambio drástico en el mapa de memoria del hijo, manteniendo el del padre inmutado. El mapa de memoria anterior fue completamente descartado y sustituido por uno nuevo, y por eso mismo no hace falta liberar expresamente la memoria reservada mediante el *malloc*. El segmento de código original, respaldado por el binario main4 en el rango 0x633c2479e000, desapareció y fue reemplazado por las regiones correspondientes al nuevo ejecutable programa_hijo, cargado en una dirección base distinta (0x616e3dea3000). Además, se observó una reubicación completa de las bibliotecas dinámicas (como libc.so.6) y de la pila.

```
616e3dea3000-616e3dea4000 r--p 00000000 103:06 306612 /home/san/programa_hijo
616e3dea4000-616e3dea5000 r-xp 00001000 103:06 306612 /home/san/programa_hijo
616e3dea5000-616e3dea6000 r--p 00002000 103:06 306612 /home/san/programa_hijo
616e3dea6000-616e3dea7000 r--p 00002000 103:06 306612 /home/san/programa_hijo
616e3dea7000-616e3dea8000 rw-p 00003000 103:06 306612 /home/san/programa_hijo
616e6fb97000-616e6fbb8000 rw-p 00000000 00:00 0 [heap]
7f6eb3e00000-7f6eb3e28000 r--p 00000000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
7f6eb3e28000-7f6eb3fb0000 r-xp 00028000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
7f6eb3fb0000-7f6eb3fff000 r--p 001b0000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
7f6eb3fff000-7f6eb4003000 r--p 001fe000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
7f6eb4003000-7f6eb4005000 rw-p 00202000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
7f6eb4005000-7f6eb4012000 rw-p 00000000 00:00 0
7f6eb4202000-7f6eb4205000 rw-p 00000000 00:00 0
7f6eb421d000-7f6eb421f000 rw-p 00000000 00:00 0
7f6eb421f000-7f6eb4221000 r--p 00000000 00:00 0 [vvar]
7f6eb4221000-7f6eb4223000 r--p 00000000 00:00 0 [vvar_vclock]
7f6eb4223000-7f6eb4225000 r-xp 00000000 00:00 0 [vdso]
7f6eb4225000-7f6eb4226000 r--p 00000000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7f6eb4226000-7f6eb4251000 r-xp 00001000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7f6eb4251000-7f6eb425b000 r--p 0002c000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7f6eb425b000-7f6eb425d000 r--p 00036000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7f6eb425d000-7f6eb425f000 rw-p 00038000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7ffc2bd6e000-7ffc2bd8f000 rw-p 00000000 00:00 0 [stack]
fffffffff600000-fffffffff601000 --xp 00000000 00:00 0 [vsyscall]
```

A pesar de esta renovación total de la memoria, donde desaparecieron tanto el código antiguo como las variables del heap reservadas anteriormente, el identificador del proceso (PID 168524) permaneció inalterado. Esto demuestra que *exec* no crea un proceso nuevo, sino que recicla el proceso existente vaciando su memoria y cargando en ella un nuevo programa desde el disco, pero conservando sus atributos identificativos.

EJERCICIO 5:

Compilación estática

```
PID: 262693
Operación con math.h (sin + pow) sobre 1.00 = 1.841471
Pulsa ENTER para terminar...
```

Compilación dinámica

```
PID: 265258
Operación con math.h (sin + pow) sobre 1.00 = 1.841471
Pulsa ENTER para terminar...
```

En este apartado se ha analizado cómo influye el tipo de enlace (estático o dinámico) en el mapa de memoria de un proceso. Se hizo un programa que dependiese de la librería matemática (libm), realizando operaciones de seno y potencia. Se generaron dos ejecutables a partir del mismo código fuente: uno mediante compilación estándar con enlace dinámico (PID 262693) y otro forzando el enlace estático (PID 265258).

El análisis del mapa de memoria correspondiente a la versión con enlace dinámico revela un espacio de direcciones altamente fragmentado. Además de las regiones propias del ejecutable main5, el sistema operativo ha mapeado explícitamente múltiples archivos externos necesarios para la ejecución. Se pueden distinguir claramente las direcciones de memoria asignadas a la biblioteca matemática compartida (libm.so.6), a la biblioteca estándar de C (libc.so.6) y al cargador dinámico (ld-linux-x86-64.so.2). Como consecuencia de esta delegación de funciones, el segmento de texto del ejecutable principal es extremadamente reducido, ocupando apenas 4 KB (una página de memoria), ya que la mayor parte de la lógica compleja reside en las librerías compartidas externas.

```
5a9861133000-5a9861134000 r--p 00000000 103:06 306129 /home/san/main5
5a9861134000-5a9861135000 r-xp 00001000 103:06 306129 /home/san/main5
5a9861135000-5a9861136000 r--p 00002000 103:06 306129 /home/san/main5
5a9861136000-5a9861137000 r--p 00002000 103:06 306129 /home/san/main5
5a9861137000-5a9861138000 rw-p 00003000 103:06 306129 /home/san/main5
5a9886e0e000-5a9886e2f000 rw-p 00000000 00:00 0 [heap]
7adaff800000-7adaff828000 r--p 00000000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
7adaff828000-7adaff9b0000 r-xp 00028000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
7adaff9b0000-7adaff9ff000 r--p 001b0000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
7adaff9ff000-7adaffa03000 r--p 001fe000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
7adaffa03000-7adaffa05000 rw-p 00202000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
7adaffa05000-7adaffa12000 rw-p 00000000 00:00 0
7adaffbaf000-7adaffbb2000 rw-p 00000000 00:00 0
7adaffbb2000-7adaffbc2000 r--p 00000000 103:06 5542911 /usr/lib/x86_64-linux-gnu/libm.so.6
7adaffbc2000-7adaffc41000 r-xp 00010000 103:06 5542911 /usr/lib/x86_64-linux-gnu/libm.so.6
7adaffc41000-7adaffc99000 r--p 0008f000 103:06 5542911 /usr/lib/x86_64-linux-gnu/libm.so.6
7adaffc99000-7adaffc9a000 r--p 000e7000 103:06 5542911 /usr/lib/x86_64-linux-gnu/libm.so.6
7adaffc9a000-7adaffc9b000 rw-p 000e8000 103:06 5542911 /usr/lib/x86_64-linux-gnu/libm.so.6
7adaffcb3000-7adaffcb5000 rw-p 00000000 00:00 0
7adaffcb5000-7adaffcb7000 r--p 00000000 00:00 0 [vvar]
7adaffcb7000-7adaffcb9000 r--p 00000000 00:00 0 [vvar_vclock]
7adaffcb9000-7adaffcbb000 r-xp 00000000 00:00 0 [vdso]
7adaffcbb000-7adaffcbc000 r--p 00000000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7adaffcbc000-7adaffce7000 r-xp 00001000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7adaffce7000-7adaffcf1000 r--p 0002c000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7adaffcf1000-7adaffcf3000 r--p 00036000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7adaffcf3000-7adaffcf5000 rw-p 00038000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7ffdf5196000-7ffdf51b7000 rw-p 00000000 00:00 0 [stack]
fffffffff600000-fffffffff601000 --xp 00000000 00:00 0 [vsyscall]
```

Por el contrario, el mapa de memoria de la versión con enlace estático presenta una estructura radicalmente distinta y simplificada. En este caso, han desaparecido por completo las referencias a bibliotecas compartidas externas; no

hay rastro de libm, libc ni del cargador dinámico en el listado de regiones mapeadas. Sin embargo, esta aparente simplicidad en el mapa conlleva un cambio drástico en el tamaño del propio ejecutable. El segmento de código (r-xp) de la compilación estática muestra un crecimiento masivo, pasando de los 4 KB de la versión dinámica a aproximadamente 667 KB.

```
00400000-00401000 r--p 00000000 103:06 306794 /home/san/main5_statico
00401000-004a4000 r-xp 00001000 103:06 306794 /home/san/main5_statico
004a4000-004ce000 r--p 000a4000 103:06 306794 /home/san/main5_statico
004ce000-004d3000 r--p 000cd000 103:06 306794 /home/san/main5_statico
004d3000-004d5000 rw-p 000d2000 103:06 306794 /home/san/main5_statico
004d5000-004db000 rw-p 00000000 00:00 0
0a3df000-0a401000 rw-p 00000000 00:00 0 [heap]
71cecc6d4000-71cecc6d6000 r--p 00000000 00:00 0 [vvar]
71cecc6d6000-71cecc6d8000 r--p 00000000 00:00 0 [vvar_vclock]
71cecc6d8000-71cecc6da000 r-xp 00000000 00:00 0 [vdso]
7fff3c2d5000-7fff3c2f6000 rw-p 00000000 00:00 0 [stack]
ffffffff600000-ffffffff601000 --xp 00000000 00:00 0 [vsyscall]
```

Este aumento de tamaño confirma que, durante el proceso de enlace estático, el compilador ha extraído el código máquina necesario de las bibliotecas (incluyendo las funciones sin, pow y la gestión de entrada/salida de printf) y lo ha copiado físicamente dentro del segmento de texto del nuevo ejecutable. El resultado es un proceso totalmente autocontenido que no requiere cargar dependencias externas en tiempo de ejecución, a costa de incrementar significativamente el consumo de memoria privada para almacenar código que, en el modelo dinámico, podría ser compartido por múltiples procesos.

EJERCICIO 6:

```
PID: 478758
Dirección de la variable global en el hilo principal: 0x555838578010
Dirección de la variable local en el hilo principal: 0x7ffd5d5cd0fc

Hilo 0 - Dirección de la variable global en el hilo: 0x555838578010
Hilo 0 - Dirección del parámetro recibido en el hilo: 0x7ffd5d5cd108
Hilo 0 - Dirección de la variable local en el hilo: 0x70b91cdfeea8
Hilo 1 - Dirección de la variable global en el hilo: 0x555838578010
Hilo 1 - Dirección del parámetro recibido en el hilo: 0x7ffd5d5cd10c
Hilo 1 - Dirección de la variable local en el hilo: 0x70b91c5fdea8
Dirección de la memoria dinámica asignada en el hilo: 0x70b914000b70
Dirección de la memoria dinámica asignada en el hilo: 0x70b90c000b70
Hilo 0 - Esperando a que el usuario pulse ENTER en el main...
Hilo 1 - Esperando a que el usuario pulse ENTER en el main...

Todos los hilos han terminado. ENTER para fin del programa.
```

En este último apartado se ha analizado la estructura de memoria de un proceso multihilo (PID 478758), observando cómo comparten recursos globales mientras mantienen espacios privados para su ejecución. La ejecución del programa generó un hilo principal y dos hilos secundarios, permitiendo contrastar la ubicación de sus variables en el mapa de

memoria.

El análisis de la variable global var_global confirmó la naturaleza compartida del espacio de direcciones en un proceso multihilo. Tanto el hilo principal como los dos hilos secundarios accedieron a esta variable en la misma dirección de memoria (0x555838578010), ubicada en el segmento de datos (rw-p) respaldado por el propio ejecutable main6. Esto corrobora que las secciones de datos y código son comunes para todos los hilos de ejecución dentro del proceso.

Antes de crear los hilos

```
555838574000-555838575000 r--p 00000000 103:06 306839 /home/san/main6
555838575000-555838576000 r-xp 00001000 103:06 306839 /home/san/main6
555838576000-555838577000 r--p 00002000 103:06 306839 /home/san/main6
555838577000-555838578000 r--p 00002000 103:06 306839 /home/san/main6
555838578000-555838579000 rw-p 00003000 103:06 306839 /home/san/main6
55585d992000-55585d9b3000 rw-p 00000000 00:00 0 [heap]
70b91ce00000-70b91ce28000 r--p 00000000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
70b91ce28000-70b91cfb0000 r-xp 00028000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
70b91cfb0000-70b91cfff000 r--p 001b0000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
70b91cfff000-70b91d003000 r--p 001fe000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
70b91d003000-70b91d005000 rw-p 00202000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
70b91d005000-70b91d012000 rw-p 00000000 00:00 0
70b91d126000-70b91d129000 rw-p 00000000 00:00 0
70b91d141000-70b91d143000 rw-p 00000000 00:00 0
70b91d143000-70b91d145000 r--p 00000000 00:00 0 [vvar]
70b91d145000-70b91d147000 r--p 00000000 00:00 0 [vvar_vclock]
70b91d147000-70b91d149000 r-xp 00000000 00:00 0 [vdso]
70b91d149000-70b91d14a000 r--p 00000000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
70b91d14a000-70b91d175000 r-xp 00001000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
70b91d175000-70b91d17f000 r--p 0002c000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
70b91d17f000-70b91d181000 r--p 00036000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
70b91d181000-70b91d183000 rw-p 00038000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7ffd5d5af000-7ffd5d5d0000 rw-p 00000000 00:00 0 [stack]
fffffffff600000-fffffffff601000 --xp 00000000 00:00 0 [vsyscall]
```

Después de crear los hilos y que estos reserven memoria dinámica

```
555838574000-555838575000 r--p 00000000 103:06 306839 /home/san/main6
555838575000-555838576000 r-xp 00001000 103:06 306839 /home/san/main6
555838576000-555838577000 r--p 00002000 103:06 306839 /home/san/main6
555838577000-555838578000 r--p 00002000 103:06 306839 /home/san/main6
555838578000-555838579000 rw-p 00003000 103:06 306839 /home/san/main6
55585d992000-55585d9b3000 rw-p 00000000 00:00 0 [heap]
70b90c000000-70b90c021000 rw-p 00000000 00:00 0
70b90c021000-70b910000000 ---p 00000000 00:00 0
70b914000000-70b914021000 rw-p 00000000 00:00 0
70b914021000-70b918000000 ---p 00000000 00:00 0
70b91bdf0000-70b91bdf0000 ---p 00000000 00:00 0
70b91bdf0000-70b91c5ff000 rw-p 00000000 00:00 0
70b91c5ff000-70b91c600000 ---p 00000000 00:00 0
70b91c600000-70b91ce00000 rw-p 00000000 00:00 0
70b91ce00000-70b91ce28000 r--p 00000000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
70b91ce28000-70b91cfb0000 r-xp 00028000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
70b91cfb0000-70b91cfff000 r--p 001b0000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
70b91cfff000-70b91d003000 r--p 001fe000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
70b91d003000-70b91d005000 rw-p 00202000 103:06 5542908 /usr/lib/x86_64-linux-gnu/libc.so.6
70b91d005000-70b91d012000 rw-p 00000000 00:00 0
70b91d126000-70b91d129000 rw-p 00000000 00:00 0
70b91d141000-70b91d143000 rw-p 00000000 00:00 0
70b91d143000-70b91d145000 r--p 00000000 00:00 0 [vvar]
70b91d145000-70b91d147000 r--p 00000000 00:00 0 [vvar_vclock]
70b91d147000-70b91d149000 r-xp 00000000 00:00 0 [vdso]
70b91d149000-70b91d14a000 r--p 00000000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
70b91d14a000-70b91d175000 r-xp 00001000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
70b91d175000-70b91d17f000 r--p 0002c000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
70b91d17f000-70b91d181000 r--p 00036000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
70b91d181000-70b91d183000 rw-p 00038000 103:06 5542905 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7ffd5d5af000-7ffd5d5d0000 rw-p 00000000 00:00 0 [stack]
fffffffff600000-fffffffff601000 --xp 00000000 00:00 0 [vsyscall]
```

Por el contrario, el estudio de las variables locales reveló la existencia de pilas independientes. Mientras que la variable local del hilo principal se alojó en la región tradicional etiquetada como [stack] en direcciones altas (0x7ffd5d5...), las variables locales de los hilos secundarios (var_local_hilo) se ubicaron en direcciones distintas (0x70b91c...). Al examinar el mapa de memoria, se identificaron dos nuevas regiones de memoria de 8 MB cada una (por ejemplo, el rango 70b91c600000-70b91ce00000), creadas específicamente para servir como pila de ejecución para cada uno de los nuevos hilos.

Después de que desaparezcan los hilos

555838574000-555838575000	r--p	00000000	103:06	306839	/home/san/main6
555838575000-555838576000	r-xp	00001000	103:06	306839	/home/san/main6
555838576000-555838577000	r--p	00002000	103:06	306839	/home/san/main6
555838577000-555838578000	r--p	00002000	103:06	306839	/home/san/main6
555838578000-555838579000	rw-p	00003000	103:06	306839	/home/san/main6
55585d992000-55585d9b3000	rw-p	00000000	00:00	0	[heap]
70b90c000000-70b90c021000	rw-p	00000000	00:00	0	
70b90c021000-70b910000000	--p	00000000	00:00	0	
70b914000000-70b914021000	rw-p	00000000	00:00	0	
70b914021000-70b918000000	--p	00000000	00:00	0	
70b91bdf0000-70b91bdf0000	--p	00000000	00:00	0	
70b91bdf0000-70b91c5ff000	rw-p	00000000	00:00	0	
70b91c5ff000-70b91c600000	--p	00000000	00:00	0	
70b91c600000-70b91ce00000	rw-p	00000000	00:00	0	
70b91ce00000-70b91ce28000	r--p	00000000	103:06	5542908	/usr/lib/x86_64-linux-gnu/libc.so.6
70b91ce28000-70b91cfb0000	r-xp	00028000	103:06	5542908	/usr/lib/x86_64-linux-gnu/libc.so.6
70b91cfb0000-70b91cfff000	r--p	001b0000	103:06	5542908	/usr/lib/x86_64-linux-gnu/libc.so.6
70b91cfff000-70b91d003000	r--p	001fe000	103:06	5542908	/usr/lib/x86_64-linux-gnu/libc.so.6
70b91d003000-70b91d005000	rw-p	00202000	103:06	5542908	/usr/lib/x86_64-linux-gnu/libc.so.6
70b91d005000-70b91d012000	rw-p	00000000	00:00	0	
70b91d0f8000-70b91d0fc000	r--p	00000000	103:06	5516652	/usr/lib/x86_64-linux-gnu/libgcc_s.so.1
70b91d0fc000-70b91d120000	r-xp	00004000	103:06	5516652	/usr/lib/x86_64-linux-gnu/libgcc_s.so.1
70b91d120000-70b91d124000	r--p	00028000	103:06	5516652	/usr/lib/x86_64-linux-gnu/libgcc_s.so.1
70b91d124000-70b91d125000	r--p	0002b000	103:06	5516652	/usr/lib/x86_64-linux-gnu/libgcc_s.so.1
70b91d125000-70b91d126000	rw-p	0002c000	103:06	5516652	/usr/lib/x86_64-linux-gnu/libgcc_s.so.1
70b91d126000-70b91d129000	rw-p	00000000	00:00	0	
70b91d141000-70b91d143000	rw-p	00000000	00:00	0	
70b91d143000-70b91d145000	r--p	00000000	00:00	0	[vvar]
70b91d145000-70b91d147000	r--p	00000000	00:00	0	[vvar_vclock]
70b91d147000-70b91d149000	r-xp	00000000	00:00	0	[vdso]
70b91d149000-70b91d14a000	r--p	00000000	103:06	5542905	/usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
70b91d14a000-70b91d175000	r-xp	00001000	103:06	5542905	/usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
70b91d175000-70b91d17f000	r--p	0002c000	103:06	5542905	/usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
70b91d17f000-70b91d181000	r--p	00036000	103:06	5542905	/usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
70b91d181000-70b91d183000	rw-p	00038000	103:06	5542905	/usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7ffd5d5af000-7ffd5d5d0000	rw-p	00000000	00:00	0	[stack]
fffffffff600000-fffffffff601000	--xp	00000000	00:00	0	[vsyscall]

Finalmente, las direcciones devueltas por *malloc* dentro de los hilos secundarios (0x70b914000b70 y 0x70b90c000b70) no se ubicaron en el segmento [heap] principal del proceso (0x5558...), sino en nuevas regiones de memoria.