

# Práctica 5

Jorge García y Antón Jaureguizar

## Ejercicio 1:

### EJECUCIÓN:

```
El PID del proceso es: 9296
Dirección de la función main: 0x557383ddf1c9
Dirección de var_global1: 0x557383de2010
Dirección de var_global2: 0x557383de2014
Dirección de var_global3: 0x557383de2038
Dirección de array_global: 0x557383de2020
Dirección de var_local1: 0x7ffd2e66238
Dirección de var_local2: 0x7ffd2e66248
Dirección de array_3d: 0x7ffd2e66250

Orden de almacenamiento del array 3D en memoria:
0x7ffd2e66250
0x7ffd2e66254
0x7ffd2e66278
```

### MAPA DE MEMORIA:

```
jorge@jorge-14Z900-G-AA748:~/Documentos/SOI_P5$ cat /proc/9296/maps
557383dde000-557383ddf000 r--p 00000000 103:07 4985372
557383ddf000-557383de0000 r--p 00001000 103:07 4985372
557383de0000-557383de1000 r--p 00002000 103:07 4985372
557383de1000-557383de2000 r--p 00002000 103:07 4985372
557383de2000-557383de3000 rw-p 00003000 103:07 4985372
557385529000-55738554a000 rw-p 00000000 00:00 0
[heap]
7e6009228000-7e6009228000 r--p 00000000 103:07 4467942
7e6009228000-7e60093b0000 r--p 00028000 103:07 4467942
7e60093b0000-7e60093ff000 r--p 001b0000 103:07 4467942
7e60093ff000-7e6009403000 r--p 001fe000 103:07 4467942
7e6009403000-7e6009405000 rw-p 00202000 103:07 4467942
7e6009405000-7e6009412000 rw-p 00000000 00:00 0
7e6009530000-7e6009533000 rw-p 00000000 00:00 0
7e6009541000-7e6009543000 rw-p 00000000 00:00 0
7e6009543000-7e6009545000 r--p 00000000 00:00 0
7e6009545000-7e6009547000 r--p 00000000 00:00 0
7e6009547000-7e6009549000 r--p 00000000 00:00 0
7e6009549000-7e600954a000 r--p 00000000 103:07 4467760
7e600954a000-7e6009575000 r--p 00001000 103:07 4467760
7e6009575000-7e600957f000 r--p 0002c000 103:07 4467760
7e600957f000-7e6009581000 r--p 00036000 103:07 4467760
7e6009581000-7e6009583000 rw-p 00038000 103:07 4467760
7ffd2e48000-7ffd2e69000 rw-p 00000000 00:00 0
[stack]
ffffffffff600000-ffffffffff601000 --xp 00000000 00:00 0
[vsyscall]
```

El main se encuentra en la sección .text del ejecutable. Lo podemos identificar porque en la columna de protección tiene la x de “executable”. En concreto se encuentra en “557383ddf000-557383de0000 r--p 00001000 103:07 4985372” /home/jorge/Documentos/SOI\_P5/a”

Las variables globales se encuentran en la sección .data del ejecutable. Se puede identificar dentro del ejecutable porque en la columna de protección tiene rw, que indica que se puede leer y escribir en estas variables. En concreto se encuentran en:

“557383de2000-557383de3000 rw-p 00003000 103:07 4985372” /home/jorge/Documentos/SOI\_P5/a”

Además, hay una variable global sin inicializar que se encuentra en la sección contigua a las inicializadas. Como se puede observar, en una primera ejecución var\_global3 estaba inicializada y la dirección de memoria era la inmediatamente contigua a var\_global2. Más tarde, al dejarla sin inicializar cambia ligeramente esta dirección al ubicarse en otra zona diferente del mapa de memoria.

Las variables locales se encuentran en la pila (stack). Al igual que las globales, las podemos identificar por el rw en la columna de protección. Se encuentran en:

"7ffdd2e48000-7ffdd2e69000 rw-p 00000000 00:00 0 [stack]"

Por último, como podemos ver gracias a los printf, el array 3D se almacena de manera consecutiva siguiendo el último índice. Esto lo podemos ver ya que array\_3d[0][0][0] y array\_3d[0][0][1] tienen posiciones de memoria consecutivas.

## Ejercicio 2:

EJECUCIÓN:

```
PID: 10114
Funcion f1:
Direccion de parametro: 0x7fffff476443c
Direccion de variable local: 0x7fffff4764444
Funcion f2:
Direccion de parametro: 0x7fffff476443c
Direccion de variable local: 0x7fffff4764444
```

MAPA DE MEMORIA:

6006696a0000-6006696a1000 r--p 00000000 103:05 2915179	/home/anton/vscode/SOI/Practica5/SOI_P5/ej2
6006696a1000-6006696a2000 r-xp 00001000 103:05 2915179	/home/anton/vscode/SOI/Practica5/SOI_P5/ej2
6006696a2000-6006696a3000 r--p 00002000 103:05 2915179	/home/anton/vscode/SOI/Practica5/SOI_P5/ej2
6006696a3000-6006696a4000 r--p 00002000 103:05 2915179	/home/anton/vscode/SOI/Practica5/SOI_P5/ej2
6006696a4000-6006696a5000 rw-p 00003000 103:05 2915179	/home/anton/vscode/SOI/Practica5/SOI_P5/ej2
6006a3bbb000-6006a3bcd000 rw-p 00000000 00:00 0	[heap]
7b6e93c00000-7b6e93c28000 r--p 00000000 103:05 4066780	/usr/lib/x86_64-linux-gnu/libc.so.6
7b6e93c28000-7b6e93db000 r-xp 00028000 103:05 4066780	/usr/lib/x86_64-linux-gnu/libc.so.6
7b6e93db000-7b6e93e15000 r--p 001bd000 103:05 4066780	/usr/lib/x86_64-linux-gnu/libc.so.6
7b6e93e15000-7b6e93e16000 ---p 00215000 103:05 4066780	/usr/lib/x86_64-linux-gnu/libc.so.6
7b6e93e16000-7b6e93e1a000 r--p 00215000 103:05 4066780	/usr/lib/x86_64-linux-gnu/libc.so.6
7b6e93e1a000-7b6e93e1c000 rw-p 00219000 103:05 4066780	/usr/lib/x86_64-linux-gnu/libc.so.6
7b6e93e1c000-7b6e93e29000 rw-p 00000000 00:00 0	/usr/lib/x86_64-linux-gnu/libc.so.6
7b6e94013000-7b6e94016000 rw-p 00000000 00:00 0	
7b6e94025000-7b6e94027000 rw-p 00000000 00:00 0	
7b6e94027000-7b6e94029000 r--p 00000000 103:05 4066764	/usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7b6e94029000-7b6e94053000 r-xp 00002000 103:05 4066764	/usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7b6e94053000-7b6e9405e000 r--p 0002c000 103:05 4066764	/usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7b6e9405f000-7b6e94061000 r--p 00037000 103:05 4066764	/usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7b6e94061000-7b6e94063000 rw-p 00039000 103:05 4066764	/usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7ffff4745000-7ffff4766000 rw-p 00000000 00:00 0	[stack]
7ffff47a1000-7ffff47a5000 r--p 00000000 00:00 0	[vvar]
7ffff47a5000-7ffff47a7000 r-xp 00000000 00:00 0	[vdso]
ffffffffffff600000-ffffffffffff601000 --xp 00000000 00:00 0	[vsyscall]

Los códigos de la funciones f1() y f2() se encuentran en el sección .text junto al main(), que podemos identificar gracias al valor ejecutable (r-xp):

"6006696a1000-6006696a2000 r-xp 00001000 103:05 2915179/home/anton/vscode/SOI/Practica5/SOI\_P5/ej2"

Podemos ver en la ejecución del código que ambas funciones otorgan la misma dirección de memoria al parámetro recibido y a la variable local, esto es porque las variables locales de funciones se guardan en pila [stack]:

"7ffff4745000-7ffff4766000 rw-p 00000000 00:00 0 [stack]"

Al ejecutarse la función f1() se guardan tanto el parámetro como la variable local creada, y al acabar esta función la pila se vacía, es decir; se liberan las direcciones de memoria ocupadas durante la ejecución. Al ejecutarse f2(), el puntero a pila inicia en el mismo sitio donde inició en f1() haciendo que tanto el parámetro como la variable local se guarden en la misma dirección de memoria que f1().

# Ejercicio 3:

## EJECUCIÓN:

```
El PID del proceso es: 10763
Se reserva poca memoria con un malloc
Dirección devuelta por malloc1: 0x574f385e3ac0

Se reserva mucha memoria con un malloc
Dirección devuelta por malloc2: 0x7ac96dffff010

Se hacen dos llamadas a malloc de diferentes tamaños
Dirección devuelta por malloc3: 0x574f385e3b50
Dirección devuelta por malloc4: 0x574f385e3c60

Se reserva memoria con alloca
Dirección devuelta por alloca: 0x7ffffe5605720

Se reserva memoria con malloc y luego se hace un realloc
Dirección devuelta por malloc5: 0x574f385e4070

Dirección devuelta por realloc: 0x574f385e4070

Dirección devuelta por realloc2: 0x7ac96d5fe010

Se reserva espacio para 1000 doubles con malloc
Dirección devuelta por malloc_doubles: 0x574f385e4070
El tamaño del puntero devuelto por malloc es: 8 bytes

Se cambia la dirección del puntero antes de llamar a free
Nueva dirección del puntero malloc_doubles: 0x574f385e5010
```

## MAPA DE MEMORIA AL COMIENZO

```
jorge@jorge-14Z900-G-AA74B:~$ cat /proc/10763/maps
574f32212000-574f32213000 r--p 00000000 103:07 4985287
574f32213000-574f32214000 r--xp 00001000 103:07 4985287
574f32214000-574f32215000 r--p 00002000 103:07 4985287
574f32215000-574f32216000 r--p 00003000 103:07 4985287
574f32216000-574f32217000 rw-p 00004000 103:07 4985287
574f385e3000-574f38604000 rw-p 00000000 00:00 0
7ac96ea00000-7ac96ea28000 r--p 00000000 103:07 4467942
7ac96ea28000-7ac96eb0000 r--xp 00028000 103:07 4467942
7ac96eb0000-7ac96ebff000 r--p 001b0000 103:07 4467942
7ac96ebff000-7ac96ec03000 r--p 001fe000 103:07 4467942
7ac96ec03000-7ac96ec05000 rw-p 00202000 103:07 4467942
7ac96ec05000-7ac96ec12000 rw-p 00000000 00:00 0
7ac96ed01000-7ac96ed4000 rw-p 00000000 00:00 0
7ac96ed04000-7ac96edf4000 rw-p 00000000 00:00 0
7ac96edf2000-7ac96edf6000 r--p 00000000 00:00 0
7ac96edf6000-7ac96edf8000 r--p 00000000 00:00 0
7ac96edf8000-7ac96edfa000 r--xp 00000000 00:00 0
7ac96edfa000-7ac96edfb000 r--p 00000000 103:07 4467760
7ac96edfb000-7ac96ee26000 r--xp 00001000 103:07 4467760
7ac96ee26000-7ac96ee30000 r--p 0002c000 103:07 4467760
7ac96ee30000-7ac96ee32000 r--p 00036000 103:07 4467760
7ac96ee32000-7ac96ee34000 rw-p 00038000 103:07 4467760
7ffe55e6000-7ffe5607000 rw-p 00000000 00:00 0
fffffffff600000-ffffffffffff601000 --xp 00000000 00:00 0
/home/jorge/Documentos/SOI_P5/a
/home/jorge/Documentos/SOI_P5/a
/home/jorge/Documentos/SOI_P5/a
/home/jorge/Documentos/SOI_P5/a
/home/jorge/Documentos/SOI_P5/a
[heap]
/usr/lib/x86_64-linux-gnu/libc.so.6
/usr/lib/x86_64-linux-gnu/libc.so.6
/usr/lib/x86_64-linux-gnu/libc.so.6
/usr/lib/x86_64-linux-gnu/libc.so.6
/usr/lib/x86_64-linux-gnu/libc.so.6
[vvar]
[vvar_clock]
[vdso]
/usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
/usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
/usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
/usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
/usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
[stack]
[vsyscall]
```

Al realizar un malloc pequeño se almacena en el montón, que es el espacio designado por defecto. En el mapa de memoria no aparece una nueva región porque se reutiliza el área ya existente del montón.

Cuando se ejecuta un malloc de gran tamaño (10MB) no se aloja directamente en el montón sino que se ejecuta una llamada al sistema para crear una región separada para alojar mallocs de gran tamaño. Como podemos observar, el malloc2 cae dentro de la región para estos archivos. Lo podemos diferenciar fácilmente del montón normal ya que sus direcciones de memoria empiezan por 7 en vez de por 5.

```
574f385e3000-574f38604000 rw-p 00000000 00:00 0
7ac96dff000-7ac96ea0000 rw-p 00000000 00:00 0
7ac96ea00000-7ac96ea28000 r--p 00000000 103:07 4467942
[heap]
/usr/lib/x86_64-linux-gnu/libc.so.6
```

Al ejecutar los dos malloc seguidos, al ser los dos pequeños se almacenan en el montón con direcciones de memoria contiguas, como podemos apreciar en la imagen de la ejecución del programa. Esto nos indica que están alojadas en el mismo bloque de memoria.

La función alloca almacena las variables en la pila (stack), por lo que el montón no se ve afectado. Tampoco vemos ningún cambio en la pila dentro del mapa de memoria debido a que el tamaño del malloc no es muy grande. Como podemos ver tras realizar alloca la dirección de memoria corresponde con las de la pila. Además, no se necesita hacer free de las variables ya que la pila se vacía al finalizar la ejecución del programa.

El primer malloc al ser pequeño no ocurre nada, al igual que el primero que hicimos en el programa. Tras realizar el realloc, vemos que la dirección sigue siendo la misma. Como no ampliamos muchísimo el tamaño el montón no se ve afectado y se sigue pudiendo alojar en el mismo bloque de memoria. Por último, al realizar un malloc de gran tamaño, se aloja en la sección destinada para asignaciones grandes, la misma en la que se alojó malloc2 de 10MB.

Al asignar memoria para 1000 dobles podemos ver que el montón tiene espacio suficiente para alojarlos por lo que no vemos cambios. Entonces cambiamos la dirección del puntero aumentándole en 500 doubles. En ese momento, si hiciésemos un free daría error, por lo que debemos guardar el puntero original. Tampoco ocurren cambios en el mapa de memoria.

Si hacemos sizeof del puntero nos devolverá siempre 8 bytes, que es lo que ocupa una dirección de memoria en esta arquitectura.

## Ejercicio 4:

Al crear un proceso hijo podemos ver que los mapas de memoria son idénticos. Esto tiene sentido ya que comparten variables, librerías y código.

Al realizar una asignación de memoria dinámica mediante un malloc en el proceso hijo vemos que ocurren cambios únicamente en el mapa de memoria del hijo, quedando el del padre invariante. Esto se debe a que las asignaciones dinámicas que se realizan desde un proceso hijo son independientes a su padre, ya que cada uno cuenta con su propio montón. Para ver el cambio hemos realizado un malloc de gran tamaño (10MB) para que se crease una sección nueva para alojarlo.

Por último, al ejecutar execv se reemplaza el proceso actual y podemos ver que el mapa de memoria cambia completamente. Se actualizan todas las direcciones de memoria y se pierde el acceso al malloc realizado antes. Esto es lógico ya que la imagen de trabajo pertenece a otro proceso iniciado mediante la llamada a execv.

### EJECUCIÓN:

```
anton@anton-Vivobook-ASUSLaptop-M1502YA-M1502YA:~/vscode/SOI/Practica5/SOI_P5$ ./ej4
El PID del proceso padre es: 5336
El PID del proceso hijo es: 5337

Mapa de memoria antes de malloc.

Se reserva memoria con malloc
Dirección devuelta por malloc: 0x700549dff010

Mapa de memoria despues de malloc.

Cambiando la imagen del proceso con execv
NUEVO PROGRAMA: PID=5337
Mapa de memoria después de execv.
```

## MAPAS DE MEMORIA ANTES DE MALLOC:

```
anton@anton-Vivobook-ASUSLaptop-M1502YA-M1502YA:~$ cat /proc/5336/maps
55a5a774c000-55a5a774d000 r--p 00000000 103:05 2887070
55a5a774d000-55a5a774e000 r-xp 00001000 103:05 2887070
55a5a774e000-55a5a774f000 r--p 00002000 103:05 2887070
55a5a774f000-55a5a7750000 r--p 00002000 103:05 2887070
55a5a7750000-55a5a7751000 rw-p 00003000 103:05 2887070
55a5c02f0000-55a5c0311000 rw-p 00000000 00:00 0
[heap]
70054a800000-70054a828000 r--p 00000000 103:05 4066780
70054a828000-70054a9bd000 r-xp 00028000 103:05 4066780
70054a9bd000-70054aa15000 r--p 001bd000 103:05 4066780
70054aa15000-70054aa16000 ---p 00215000 103:05 4066780
70054aa16000-70054aa1a000 r--p 00215000 103:05 4066780
70054aa1a000-70054aa1c000 rw-p 00219000 103:05 4066780
70054aa1c000-70054aa29000 rw-p 00000000 00:00 0
70054ab47000-70054ab4a000 rw-p 00000000 00:00 0
70054ab59000-70054ab5b000 rw-p 00000000 00:00 0
70054ab5b000-70054ab5d000 r--p 00000000 103:05 4066764
70054ab5d000-70054ab87000 r-xp 00002000 103:05 4066764
70054ab87000-70054ab92000 r--p 0002c000 103:05 4066764
70054ab93000-70054ab95000 r--p 00037000 103:05 4066764
70054ab95000-70054ab97000 rw-p 00039000 103:05 4066764
7ffd77424000-7ffd77445000 rw-p 00000000 00:00 0
7ffd774e8000-7ffd774ec000 r--p 00000000 00:00 0
7ffd774ec000-7ffd774ee000 r-xp 00000000 00:00 0
ffffffffff600000-ffffffff601000 --xp 00000000 00:00 0
anton@anton-Vivobook-ASUSLaptop-M1502YA-M1502YA:~$ cat /proc/5337/maps
55a5a774c000-55a5a774d000 r--p 00000000 103:05 2887070
55a5a774d000-55a5a774e000 r-xp 00001000 103:05 2887070
55a5a774e000-55a5a774f000 r--p 00002000 103:05 2887070
55a5a774f000-55a5a7750000 r--p 00002000 103:05 2887070
55a5a7750000-55a5a7751000 rw-p 00003000 103:05 2887070
55a5c02f0000-55a5c0311000 rw-p 00000000 00:00 0
[heap]
70054a800000-70054a828000 r--p 00000000 103:05 4066780
70054a828000-70054a9bd000 r-xp 00028000 103:05 4066780
70054a9bd000-70054aa15000 r--p 001bd000 103:05 4066780
70054aa15000-70054aa16000 ---p 00215000 103:05 4066780
70054aa16000-70054aa1a000 r--p 00215000 103:05 4066780
70054aa1a000-70054aa1c000 rw-p 00219000 103:05 4066780
70054aa1c000-70054aa29000 rw-p 00000000 00:00 0
70054ab47000-70054ab4a000 rw-p 00000000 00:00 0
70054ab59000-70054ab5b000 rw-p 00000000 00:00 0
70054ab5b000-70054ab5d000 r--p 00000000 103:05 4066764
70054ab5d000-70054ab87000 r-xp 00002000 103:05 4066764
70054ab87000-70054ab92000 r--p 0002c000 103:05 4066764
70054ab93000-70054ab95000 r--p 00037000 103:05 4066764
70054ab95000-70054ab97000 rw-p 00039000 103:05 4066764
7ffd77424000-7ffd77445000 rw-p 00000000 00:00 0
7ffd774e8000-7ffd774ec000 r--p 00000000 00:00 0
7ffd774ec000-7ffd774ee000 r-xp 00000000 00:00 0
ffffffffff600000-ffffffff601000 --xp 00000000 00:00 0
```

## MAPAS DE MEMORIA DESPUÉS DE MALLOC:

```
anton@anton-Vivobook-ASUSLaptop-M1502YA-M1502YA:~$ cat /proc/5336/maps
55a5a774c000-55a5a774d000 r--p 00000000 103:05 2887070
55a5a774d000-55a5a774e000 r-xp 00001000 103:05 2887070
55a5a774e000-55a5a774f000 r--p 00002000 103:05 2887070
55a5a774f000-55a5a7750000 r--p 00002000 103:05 2887070
55a5a7750000-55a5a7751000 rw-p 00003000 103:05 2887070
55a5c02f0000-55a5c0311000 rw-p 00000000 00:00 0
[heap]
70054a800000-70054a828000 r--p 00000000 103:05 4066780
70054a828000-70054a9bd000 r-xp 00028000 103:05 4066780
70054a9bd000-70054aa15000 r--p 001bd000 103:05 4066780
70054aa15000-70054aa16000 ---p 00215000 103:05 4066780
70054aa16000-70054aa1a000 r--p 00215000 103:05 4066780
70054aa1a000-70054aa1c000 rw-p 00219000 103:05 4066780
70054aa1c000-70054aa29000 rw-p 00000000 00:00 0
70054ab47000-70054ab4a000 rw-p 00000000 00:00 0
70054ab59000-70054ab5b000 rw-p 00000000 00:00 0
70054ab5b000-70054ab5d000 r--p 00000000 103:05 4066764
70054ab5d000-70054ab87000 r-xp 00002000 103:05 4066764
70054ab87000-70054ab92000 r--p 0002c000 103:05 4066764
70054ab93000-70054ab95000 r--p 00037000 103:05 4066764
70054ab95000-70054ab97000 rw-p 00039000 103:05 4066764
7ffd77424000-7ffd77445000 rw-p 00000000 00:00 0
7ffd774e8000-7ffd774ec000 r--p 00000000 00:00 0
7ffd774ec000-7ffd774ee000 r-xp 00000000 00:00 0
ffffffffff600000-ffffffff601000 --xp 00000000 00:00 0
```

```
anton@anton-Vivobook-ASUSLaptop-M1502YA-M1502YA:~$ cat /proc/5337/maps
55a5a774c000-55a5a774d000 r--p 00000000 103:05 2887070
55a5a774d000-55a5a774e000 r-xp 00001000 103:05 2887070
55a5a774e000-55a5a774f000 r--p 00002000 103:05 2887070
55a5a774f000-55a5a7750000 r--p 00002000 103:05 2887070
55a5a7750000-55a5a7751000 rw-p 00003000 103:05 2887070
55a5c02f0000-55a5c0311000 rw-p 00000000 00:00 0
700549dff000-70054a800000 rw-p 00000000 00:00 0
70054a800000-70054a828000 r--p 00000000 103:05 40666780
[heap]
/usr/lib/x86_64-linux-gnu/libc.so.6
```

## MAPA DE MEMORIA PROCESO HIJO DESPUÉS DE EXECL:

```
anton@anton-Vivobook-ASUSLaptop-M1502YA-M1502YA:~$ cat /proc/5337/maps
[.vsyscall]
5f0705bf8000-5f0705bf9000 r--p 00000000 103:05 2887065
5f0705bf9000-5f0705bfa000 r-xp 00001000 103:05 2887065
5f0705bfa000-5f0705bfb000 r--p 00002000 103:05 2887065
5f0705bfb000-5f0705bfc000 r--p 00002000 103:05 2887065
5f0705bfc000-5f0705bfd000 rw-p 00003000 103:05 2887065
5f073f41f000-5f073f440000 rw-p 00000000 00:00 0
[heap]
7d996ac00000-7d996ac28000 r--p 00000000 103:05 40666780
7d996ac28000-7d996adb000 r-xp 00028000 103:05 40666780
7d996adb000-7d996ae15000 r--p 001bd000 103:05 40666780
7d996ae15000-7d996ae16000 ---p 00215000 103:05 40666780
7d996ae16000-7d996ae1a000 r--p 00215000 103:05 40666780
7d996ae1a000-7d996ae1c000 rw-p 00219000 103:05 40666780
7d996ae1c000-7d996ae29000 rw-p 00000000 00:00 0
7d996aeef8000-7d996aefb000 rw-p 00000000 00:00 0
7d996af0a000-7d996af0c000 rw-p 00000000 00:00 0
7d996af0c000-7d996af0e000 r--p 00000000 103:05 40666764
7d996af0e000-7d996af38000 r-xp 00002000 103:05 40666764
7d996af38000-7d996af43000 r--p 0002c000 103:05 40666764
7d996af44000-7d996af46000 r--p 00037000 103:05 40666764
7d996af46000-7d996af48000 rw-p 00039000 103:05 40666764
7ffcf051000-7ffcf0572000 rw-p 00000000 00:00 0
[stack]
7ffcf05c5000-7ffcf05c9000 r--p 00000000 00:00 0
7ffcf05c9000-7ffcf05cb000 r-xp 00000000 00:00 0
ffffffffff600000-ffffffffff601000 --xp 00000000 00:00 0
[vvar]
[vdso]
[vsyscall]
```

## Ejercicio 5:

Al compilar el programa mediante un enlace dinámico las librerías no se guardan en el ejecutable sino que se cargan en memoria una vez comienza la ejecución del programa. Con esto se obtiene un ejecutable de menor tamaño pero el mapa de memoria ocupa más. Podemos observar varios archivos libc.so que son los que contienen la librería matemática cargados en memoria.

Por otro lado, al compilar mediante un enlace estático, el ejecutable contiene las librerías utilizadas, entre ellas la librería matemática por lo que no es necesario cargarlas en tiempo de compilación. Podemos ver entonces que el mapa de memoria es mucho más pequeño y no vemos ningún archivo libc.so o similar.

### TAMAÑOS EJECUTABLES:

```
anton@anton-Vivobook-ASUSLaptop-M1502YA-M1502YA:~/vsco
ls -lh ./dinamico
-rwxrwxr-x 1 anton anton 16K nov 19 12:54 ./dinamico
anton@anton-Vivobook-ASUSLaptop-M1502YA-M1502YA:~/vsco
ls -lh ./estatico
-rw-rw-r-x 1 anton anton 880K nov 19 12:55 ./estatico
```

### COMPILEACIÓN Y EJECUCIÓN:

```

anton@anton-Vivobook-ASUSLaptop-M1502YA-M1502YA:~/vscode/SOI/Practica5/SOI_P5$ gcc ej5.c -o dinamico -lm
anton@anton-Vivobook-ASUSLaptop-M1502YA-M1502YA:~/vscode/SOI/Practica5/SOI_P5$ ./dinamico
El PID del proceso es: 6159
0.841471

anton@anton-Vivobook-ASUSLaptop-M1502YA-M1502YA:~/vscode/SOI/Practica5/SOI_P5$ gcc ej5.c -o estatico -lm -static
anton@anton-Vivobook-ASUSLaptop-M1502YA-M1502YA:~/vscode/SOI/Practica5/SOI_P5$ ./estatico
El PID del proceso es: 6170
0.841471

```

## MAPAS DE MEMORIA:

```

anton@anton-Vivobook-ASUSLaptop-M1502YA-M1502YA:~$ cat /proc/6159/maps
55a6f989f000-55a6f98a0000 r--p 00000000 103:05 2887761
55a6f98a0000-55a6f98a1000 r--p 00001000 103:05 2887761
55a6f98a1000-55a6f98a2000 r--p 00002000 103:05 2887761
55a6f98a2000-55a6f98a3000 r--p 00002000 103:05 2887761
55a6f98a3000-55a6f98a4000 rw-p 00003000 103:05 2887761
55a717a37000-55a717a58000 rw-p 00000000 00:00 0
7204b0600000-7204b0628000 r--p 00000000 103:05 4066780
7204b0628000-7204b07bd000 r--p 00028000 103:05 4066780
7204b07bd000-7204b0815000 r--p 001bd000 103:05 4066780
7204b0815000-7204b0816000 ---p 00215000 103:05 4066780
7204b0816000-7204b081a000 r--p 00215000 103:05 4066780
7204b081a000-7204b081c000 rw-p 00219000 103:05 4066780
7204b081c000-7204b0829000 rw-p 00000000 00:00 0
7204b0944000-7204b0947000 rw-p 00000000 00:00 0
7204b0956000-7204b0958000 rw-p 00000000 00:00 0
7204b0958000-7204b095a000 r--p 00000000 103:05 4066764
7204b095a000-7204b0984000 r--p 00002000 103:05 4066764
7204b0984000-7204b098f000 r--p 0002c000 103:05 4066764
7204b0990000-7204b0992000 r--p 00037000 103:05 4066764
7204b0992000-7204b0994000 rw-p 00039000 103:05 4066764
7ffd56968000-7ffd56989000 rw-p 00000000 00:00 0
7ffd569d000-7ffd569d1000 r--p 00000000 00:00 0
7ffd569d1000-7ffd569d3000 r--p 00000000 00:00 0
ffffffffffff600000-ffffffffffff601000 --xp 00000000 00:00 0
anton@anton-Vivobook-ASUSLaptop-M1502YA-M1502YA:~$ cat /proc/6170/maps
00400000-00401000 r--p 00000000 103:05 2887465
00401000-00498000 r--p 00001000 103:05 2887465
00498000-004c1000 r--p 00098000 103:05 2887465
004c1000-004c5000 r--p 000c0000 103:05 2887465
004c5000-004c8000 rw-p 000c4000 103:05 2887465
004c8000-004cd000 rw-p 00000000 00:00 0
196ee000-19718000 rw-p 00000000 00:00 0
7ffebba53000-7ffebba74000 rw-p 00000000 00:00 0
7ffebbb2b000-7ffebbb2f000 r--p 00000000 00:00 0
7ffebbb2f000-7ffebbb31000 r--p 00000000 00:00 0
ffffffffffff600000-ffffffffffff601000 --xp 00000000 00:00 0

```

## Ejercicio 6:

Miramos el mapa de memoria antes de la creación de los dos hilos. Podemos ver que la variable global se encuentra en el .data del ejecutable:

5871463c5000-5871463c6000 rw-p 00003000 103:07 4984182 /home/jorge/Documentos/SOI\_P5/ej6

Y la variable local se encuentra en stack.

Tras la creación de los dos hilos, volvemos a mirar el mapa de memoria. El montón es compartido tanto por el hilo principal como por los hilos creados, mientras que cada hilo crea su propia pila privada, que podemos identificar con las dos secciones “rw-p”. El espacio que no se puede ni leer ni escribir (“---p”) es un espacio que el sistema operativo asigna para que el hilo pueda aumentar el tamaño de su pila en caso de necesitarlo.

Tanto la variable global como la variable local del main pasada como argumento mantienen sus direcciones de memoria en ambos hilos, mientras que las variables locales declaradas en cada hilo están en direcciones de memoria diferentes, cada una en su pila privada.

Al realizar un malloc grande dentro de cada uno de los hilos vemos que se crean dos secciones diferentes. La primera región rw-p es la memoria asignada para el bloque grande solicitado por malloc. Si el bloque es demasiado grande para colocarse en el heap tradicional, se reserva una región separada,, como explicamos en el ejercicio 3. Seguido de esta sección se reserva otro espacio sin permisos de lectura o escritura ("--p") para evitar solapamientos y permitir la expansión futura de la pila o del malloc. A mayores, por decisiones del sistema operativo que no son objeto de esta práctica se crea otra sección rw-p para albergar parte del malloc ejecutado.

Con el segundo malloc podemos ver que ocurre algo análogo al primero. Se crea una sección para guardar archivos de gran tamaño y se deja un espacio sin permisos para evitar solapamientos.

#### EJECUCIÓN

```
jorge@jorge-14Z90Q-G-AA74B:~/Documentos/SOI_P5$ ./ej6
El PID del proceso es: 7573
La direccion de la variable global es: 0x5871463c5010
La direccion de la variable local del main es: 0x7ffd2a5aa024

Se crean dos hilos
Valor en el hilo: 50
Direccion de la variable global en el hilo: 0x5871463c5010
Direccion de la variable recibida en el hilo: 0x7ffd2a5aa024
Direccion de la variable local en el hilo: 0x732e805feeaa8
Valor en el hilo: 50
Direccion de la variable global en el hilo: 0x5871463c5010
Direccion de la variable recibida en el hilo: 0x7ffd2a5aa024
Direccion de la variable local en el hilo: 0x732e7fdfdea8

Se reserva memoria con malloc en el hilo
Direccion devuelta por malloc en el hilo: 0x732e7ebfd010

Se reserva memoria con malloc en el hilo
Direccion devuelta por malloc en el hilo: 0x732e7e1fc010
```

MAPA ANTES DE CREAR LOS HILOS

```
jorge@jorge-14Z900-G-AA74B:~$ cat /proc/7573/maps
5871463c1000-5871463c2000 r--p 00000000 103:07 4984182
5871463c2000-5871463c3000 r-xp 00001000 103:07 4984182
5871463c3000-5871463c4000 r--p 00002000 103:07 4984182
5871463c4000-5871463c5000 r--p 00002000 103:07 4984182
5871463c5000-5871463c6000 rw-p 00003000 103:07 4984182
58716f351000-58716f372000 rw-p 00000000 00:00 0
[heap]
732e80600000-732e80628000 r--p 00000000 103:07 4467942
732e80628000-732e807b0000 r-xp 00028000 103:07 4467942
732e807b0000-732e807ff000 r--p 001b0000 103:07 4467942
732e807ff000-732e80803000 r--p 001fe000 103:07 4467942
732e80803000-732e80805000 rw-p 00202000 103:07 4467942
732e80805000-732e80812000 rw-p 00000000 00:00 0
732e808e8000-732e809eb000 rw-p 00000000 00:00 0
732e809fb000-732e809fb000 rw-p 00000000 00:00 0
732e809fb000-732e809fd000 r--p 00000000 00:00 0
732e809fd000-732e809ff000 r--p 00000000 00:00 0
732e809ff000-732e80a1000 r-xp 00000000 00:00 0
732e80a01000-732e80a02000 r--p 00000000 103:07 4467760
732e80a02000-732e80a2d000 r-xp 00001000 103:07 4467760
732e80a2d000-732e80a37000 r--p 0002c000 103:07 4467760
732e80a37000-732e80a39000 r--p 00036000 103:07 4467760
732e80a39000-732e80a3b000 rw-p 00038000 103:07 4467760
7ffd2a58c000-7ffd2a5ad000 rw-p 00000000 00:00 0
ffffffffff600000-ffffffffff601000 --xp 00000000 00:00 0
[stack]
[vsyscall]
```

## MAPA TRAS CREAR LOS HILOS

```
jorge@jorge-14Z900-G-AA74B:~$ cat /proc/7573/maps
5871463c1000-5871463c2000 r--p 00000000 103:07 4984182
5871463c2000-5871463c3000 r-xp 00001000 103:07 4984182
5871463c3000-5871463c4000 r--p 00002000 103:07 4984182
5871463c4000-5871463c5000 r--p 00002000 103:07 4984182
5871463c5000-5871463c6000 rw-p 00003000 103:07 4984182
58716f351000-58716f372000 rw-p 00000000 00:00 0
[heap]
732e7f5fe000-732e7f5ff000 ---p 00000000 00:00 0
732e7f5ff000-732e7fdfff000 rw-p 00000000 00:00 0
732e7fdfff000-732e7fe00000 ---p 00000000 00:00 0
732e7fe00000-732e80600000 rw-p 00000000 00:00 0
[stack]
```

## PRIMER MALLOC

```
jorge@jorge-14Z900-G-AA74B:~$ cat /proc/7573/maps
5871463c1000-5871463c2000 r--p 00000000 103:07 4984182
5871463c2000-5871463c3000 r-xp 00001000 103:07 4984182
5871463c3000-5871463c4000 r--p 00002000 103:07 4984182
5871463c4000-5871463c5000 r--p 00002000 103:07 4984182
5871463c5000-5871463c6000 rw-p 00003000 103:07 4984182
58716f351000-58716f372000 rw-p 00000000 00:00 0
[heap]
732e7800000-732e78021000 rw-p 00000000 00:00 0
732e78021000-732e7c000000 ---p 00000000 00:00 0
732e7ebfd000-732e7f5fe000 rw-p 00000000 00:00 0
732e7f5fe000-732e7f5ff000 ---p 00000000 00:00 0
732e7f5ff000-732e7fdfff000 rw-p 00000000 00:00 0
732e7fdfff000-732e7fe00000 ---p 00000000 00:00 0
732e7fe00000-732e80600000 rw-p 00000000 00:00 0
```

## SEGUNDO MALLOC

```
jorge@jorge-14Z900-G-AA74B:~$ cat /proc/7573/maps
5871463c1000-5871463c2000 r--p 00000000 103:07 4984182
5871463c2000-5871463c3000 r-xp 00001000 103:07 4984182
5871463c3000-5871463c4000 r--p 00002000 103:07 4984182
5871463c4000-5871463c5000 r--p 00002000 103:07 4984182
5871463c5000-5871463c6000 rw-p 00003000 103:07 4984182
58716f351000-58716f372000 rw-p 00000000 00:00 0
[heap]
732e7000000-732e70021000 rw-p 00000000 00:00 0
732e70021000-732e74000000 ---p 00000000 00:00 0
732e7800000-732e78021000 rw-p 00000000 00:00 0
732e78021000-732e7c000000 ---p 00000000 00:00 0
732e7e1fc000-732e7f5fe000 rw-p 00000000 00:00 0
732e7f5fe000-732e7f5ff000 ---p 00000000 00:00 0
732e7f5ff000-732e7fdfff000 rw-p 00000000 00:00 0
732e7fdfff000-732e7fe00000 ---p 00000000 00:00 0
732e7fe00000-732e80600000 rw-p 00000000 00:00 0
```