

Acomodo de procesos en memoria

- a. **Revisión de los archivos generados.** Como primera tarea revisé el contenido de cada uno de los programas descritos, pues en el archivo *donde_en_la_memoria.c*, están las líneas de más alto nivel que vamos a poder encontrar y las que se nos pide que identifiquemos en las secciones de memoria. Cuando se genera el *donde_en_la_memoria.s*, en este podemos encontrar algunas instrucciones en ensamblador que nos ayudan a identificar cómo se asignan algunas direcciones de memoria. Por lo que recuerdo de las clases de *Estructura de Computadoras (donde aprendimos algunos fundamentos de ensamblador)*, el programa primero se carga en la parte más alta de nuestra memoria, viéndola como el rectángulo vertical dividido en secciones al que estamos acostumbrados, luego al comenzar a ejecutar el programa se irán cargando los datos y las llamadas a las bibliotecas del sistema. Por otra parte, cuando intente generar el archivo con extensión core, ese archivo no me lo permitía crear el sistema, me hacía una observación de que debía de estar ejecutándose el proceso, así que abrí dos terminales para ejecutarlo en una y probar con el core en la otra, pero luego me daba el error que se nos mencionaba de: “ptrace operation not permitted”. Esas son mi nociones para comenzar a analizar el comportamiento que se nos pide observar.
- b. **Probar las herramientas proporcionadas.** Al utilizar el depurador de GNU (también gdb), pude observar que tiene opciones de depuración que ayudan a generar algunos procedimientos para cosas más complejas como multiprocesamiento, pues me puse a investigar un poco sobre cómo depurar un multiproceso cliente-servidor y en el cual generaban varias instancias de la terminal para cargar en uno el programa y en las otras dos terminales ir avanzando en por los *breakpoints* de cada proceso.

El punto, es que a pesar de que proporciona una visualización de la información bastante útil, me fue complicado interpretar lo que resolvía cuando ingresaba ciertos comandos. En la página de *stackoverflow* comentaban algunas cosas e intenté probar con algunas, pero me fue más conveniente luego intentar hacerlo con los archivos binarios que se generaron en las primeras fases de esta actividad, ya que con esos pues mi razonamiento de la situación se fue más a que el programa de podía dar las direcciones de los métodos y cómo se asignaba en en la parte más baja de la memoria, pues si hacíamos la visualización hexadecimal de donde en la memoria.s, tenemos que:

```

000032d8
nivermtz@nivermtz-VirtualBox:~$ hexdump -C donde_en_la_memoria.s
00000000 09 2e 66 69 6c 65 09 22 64 6f 6e 64 65 5f 65 6e |..file."donde_en|
00000010 5f 6c 61 5f 6d 65 6d 6f 72 69 61 2e 63 22 0a 09 |_la_memoria.c"..|
00000020 2e 74 65 78 74 0a 09 2e 67 6c 6f 62 6c 09 63 61 |.text...globl.ca|
00000030 64 65 6e 61 31 0a 09 2e 64 61 74 61 0a 09 2e 61 |dena1...data...a|
00000040 6c 69 67 6e 20 31 36 0a 09 2e 74 79 70 65 09 63 |lign 16...type.c|
00000050 61 64 65 6e 61 31 2c 20 40 6f 62 6a 65 63 74 0a |adena1, @object.|
00000060 09 2e 73 69 7a 65 09 63 61 64 65 6e 61 31 2c 20 |..size.cadena1, |
00000070 32 38 0a 63 61 64 65 6e 61 31 3a 0a 09 2e 73 74 |28.cadena1:...st|
00000080 72 69 6e 67 09 22 59 6f 20 73 6f 6c 6f 20 73 5c |ring."Yo solo s|
00000090 33 30 33 5c 32 35 31 20 71 75 65 20 6e 6f 20 73 |303\251 que no s|
000000a0 5c 33 30 33 5c 32 35 31 20 6e 61 64 61 22 0a 09 |\303\251 nada"..|
000000b0 2e 63 6f 6d 6d 09 63 61 64 65 6e 61 5f 74 6f 74 |.comm.cadena_tot|
000000c0 61 6c 2c 38 2c 38 0a 09 2e 67 6c 6f 62 6c 09 74 |al,8,8...globl.t|
000000d0 61 6d 61 6e 6f 0a 09 2e 61 6c 69 67 6e 20 34 0a |amano...align 4.|
000000e0 09 2e 74 79 70 65 09 74 61 6d 61 6e 6f 2c 20 40 |..type.tamano, @|
000000f0 6f 62 6a 65 63 74 0a 09 2e 73 69 7a 65 09 74 61 |object...size.ta|
00000100 6d 61 6e 6f 2c 20 34 0a 74 61 6d 61 6e 6f 3a 0a |mano, 4.tamano:.|
00000110 09 2e 6c 6f 6e 67 09 33 30 0a 09 2e 73 65 63 74 |..long.30...sect|
00000120 69 6f 6e 09 2e 72 6f 64 61 74 61 0a 2e 4c 43 30 |ion..rodata..LC0|
00000130 3a 0a 09 2e 73 74 72 69 6e 67 09 22 50 72 6f 63 |:...string."Proc|
00000140 65 73 6f 20 66 69 6c 5c 33 30 33 5c 32 36 33 73 |eso fil\303\263s|
00000150 6f 66 6f 2c 20 50 49 44 20 25 64 5c 6e 5c 6e 22 |ofo, PID %d\n\n"|
00000160 0a 2e 4c 43 31 3a 0a 09 2e 73 74 72 69 6e 67 09 |..LC1:...string.|
00000170 22 59 6f 20 73 5c 33 30 33 5c 32 36 33 6c 6f 20 |"Yo s\303\263lo ..|
00000180 73 5c 33 30 33 5c 32 35 31 20 71 75 65 20 6e 61 |s\303\251 que na|
00000190 64 61 20 73 5c 33 30 33 5c 32 35 31 22 0a 2e 4c |da s\303\251".L|
000001a0 43 32 3a 0a 09 2e 73 74 72 69 6e 67 09 22 50 65 |C2:...string."Pel

```

Utilizando hexdump con la bandera -C que nos muestra la información en caracteres alfanuméricos, podemos notar que toda la información se encuentra en regiones muy altas de la memoria, por y contiguas entonces ahí fueron a alojarse todo el texto como tal del programa original.

c. **Revisando la información dentro de nuestro ejecutable.** Cuando volvemos a aplicar el hexdump sobre nuestro archivo ejecutable, encontramos que muchas de las cadenas ya se han dispersado dentro de la memoria y que la imagen que vemos ahora es más difícil de interpretar. Así, tenemos:

i. Lo que parece ser una sección de llamada a bibliotecas del sistema, la cual si observamos la salida que nos entrega algunas de las opciones para los mapas de memoria (ingresando el PID con pmap), veríamos:

```

24029: ./donde_en_la_memoria
000055c8297ac000 4K r-x-- donde_en_la_memoria
000055c8299ad000 4K r---- donde_en_la_memoria
000055c8299ae000 4K rw--- donde_en_la_memoria
000055c82afba000 132K rw-- [ anon ]
00007f2ee1030000 1948K r-x-- libc-2.27.so
00007f2ee1217000 2048K ----- libc-2.27.so
00007f2ee1417000 16K r---- libc-2.27.so
00007f2ee141b000 8K rw--- libc-2.27.so
00007f2ee141d000 16K rw-- [ anon ]
00007f2ee1421000 156K r-x-- ld-2.27.so
00007f2ee1632000 8K rw-- [ anon ]
00007f2ee1648000 4K r---- ld-2.27.so
00007f2ee1649000 4K rw--- ld-2.27.so
00007f2ee164a000 4K rw-- [ anon ]
00007ffefc267000 132K rw-- [ pila ]
00007ffefc289000 12K r---- [ anon ]
00007ffefc28c000 4K r-x-- [ anon ]
fffffffff6000000 4K --x-- [ anon ]
total 4508K

```


Las secciones en donde pmap dice que se ejecuta el programa, están en las localidades de memoria alta, que andan en los índices de 5 hexadecimal para el bit más significativo. Mientras que sí vemos las direcciones hexadecimales del volcado que nos da el código objeto, estas puede que estén igual en memoria alta.

```

00000010 5f 6c 61 5f 6d 65 6d 6f 72 69 61 2e 63 22 0a 09 |_la_memoria.c"..|
00000020 2e 74 65 78 74 0a 09 2e 67 6c 6f 62 6c 09 63 61 |.text...globl.ca|
00000030 64 65 6e 61 31 0a 09 2e 64 61 74 61 0a 09 2e 61 |dena1...data...a|
00000040 6c 69 67 6e 20 31 36 0a 09 2e 74 79 70 65 09 63 |lign 16...type.c|
00000050 61 64 65 6e 61 31 2c 20 40 6f 62 6a 65 63 74 0a |adena1, @object.|
00000060 09 2e 73 69 7a 65 09 63 61 64 65 6e 61 31 2c 20 |..size.cadena1, |
00000070 32 38 0a 63 61 64 65 6e 61 31 3a 0a 09 2e 73 74 |28.cadena1:...st|
00000080 72 69 6e 67 09 22 59 6f 20 73 6f 6c 6f 20 73 5c |ring."Yo solo s\|
00000090 33 30 33 5c 32 35 31 20 71 75 65 20 6e 6f 20 73 |303\251 que no s|
000000a0 5c 33 30 33 5c 32 35 31 20 6e 61 64 61 22 0a 09 |\303\251 nada"..|
000000b0 2e 63 6f 6d 6d 09 63 61 64 65 6e 61 5f 74 6f 74 |.comm.cadena_tot|
000000c0 61 6c 2c 38 2c 38 0a 09 2e 67 6c 6f 62 6c 09 74 |al,8,8...globl.t|
000000d0 61 6d 61 6e 6f 0a 09 2e 61 6c 69 67 6e 20 34 0a |amano...align 4.|
000000e0 09 2e 74 79 70 65 09 74 61 6d 61 6e 6f 2c 20 40 |..type.tamano, @|
000000f0 6f 62 6a 65 63 74 0a 09 2e 73 69 7a 65 09 74 61 |object...size.ta|
00000100 6d 61 6e 6f 2c 20 34 0a 74 61 6d 61 6e 6f 3a 0a |mano, 4.tamano:.|
00000110 09 2e 6c 6f 6e 67 09 33 30 0a 09 2e 73 65 63 74 |..long.30...sect|
00000120 69 6f 6e 09 2e 72 6f 64 61 74 61 0a 2e 4c 43 30 |ion..rodata..LC0|
00000130 3a 0a 09 2e 73 74 72 69 6e 67 09 22 50 72 6f 63 |:...string."Proc|
00000140 65 73 6f 20 66 69 6c 5c 33 30 33 5c 32 36 33 73 |eso fil\303\263s|
00000150 6f 66 6f 2c 20 50 49 44 20 25 64 5c 6e 5c 6e 22 |ofo, PID %d\n\n"|
00000160 0a 2e 4c 43 31 3a 0a 09 2e 73 74 72 69 6e 67 09 |..LC1:...string.|
00000170 22 59 6f 20 73 5c 33 30 33 5c 32 36 33 6c 6f 20 |"Yo s\303\263lo |
00000180 73 5c 33 30 33 5c 32 35 31 20 71 75 65 20 6e 61 |s\303\251 que na|
00000190 64 61 20 73 5c 33 30 33 5c 32 35 31 22 0a 2e 4c |da s\303\251"...L|
000001a0 43 32 3a 0a 09 2e 73 74 72 69 6e 67 09 22 50 65 |C2:...string."Pe|
000001b0 72 6f 20 73 69 20 61 6c 67 75 69 65 6e 20 73 61 |ro si alguien sa|
000001c0 62 65 20 6d 65 6e 6f 73 22 0a 2e 4c 43 33 3a 0a |be menos"...LC3:|
000001d0 09 2e 73 74 72 69 6e 67 09 22 25 73 5c 6e 25 73 |..string."%s\n%s|

```

- ii. Por otra partes, esta sección del ejecutable que parece ser como apuntadores hacen referencia a asignaciones:

```

00000740 07 00 00 00 0d 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000750 48 83 ec 08 48 8b 05 8d 18 20 00 48 85 c0 74 02 |H...H.... .H..t.|
00000760 ff d0 48 83 c4 08 c3 00 00 00 00 00 00 00 00 00 |..H.....|
00000770 ff 35 02 18 20 00 ff 25 04 18 20 00 0f 1f 40 00 |.5.. .%.. ...@.|
00000780 ff 25 02 18 20 00 68 00 00 00 00 e9 e0 ff ff ff |.%.. .h.....|
00000790 ff 25 fa 17 20 00 68 01 00 00 00 e9 d0 ff ff ff |.%.. .h.....|
000007a0 ff 25 f2 17 20 00 68 02 00 00 00 e9 c0 ff ff ff |.%.. .h.....|
000007b0 ff 25 ea 17 20 00 68 03 00 00 00 e9 b0 ff ff ff |.%.. .h.....|
000007c0 ff 25 e2 17 20 00 68 04 00 00 00 e9 a0 ff ff ff |.%.. .h.....|
000007d0 ff 25 da 17 20 00 68 05 00 00 00 e9 90 ff ff ff |.%.. .h.....|
000007e0 ff 25 d2 17 20 00 68 06 00 00 00 e9 80 ff ff ff |.%.. .h.....|
000007f0 ff 25 ca 17 20 00 68 07 00 00 00 e9 70 ff ff ff |.%.. .h....p...|
00000800 ff 25 c2 17 20 00 68 08 00 00 00 e9 60 ff ff ff |.%.. .h....'...|
00000810 ff 25 ba 17 20 00 68 09 00 00 00 e9 50 ff ff ff |.%.. .h....P...|
00000820 ff 25 d2 17 20 00 66 90 00 00 00 00 00 00 00 00 |.%.. .f.....|
00000830 31 ed 49 89 d1 5e 48 89 e2 48 83 e4 f0 50 54 4c |1.I..^H..H...PTL|
00000840 8d 05 7a 03 00 00 48 8d 0d 03 03 00 00 48 8d 3d |...H.....H.=|
00000850 e6 00 00 00 ff 15 86 17 20 00 f4 0f 1f 44 00 00 |.....D...|
00000860 48 8d 3d c9 17 20 00 55 48 8d 05 c1 17 20 00 48 |H.=... .UH.... .H|
00000870 39 f8 48 89 e5 74 19 48 8b 05 5a 17 20 00 48 85 |9.H..t.H..Z. .H.|
00000880 c0 74 0d 5d ff e0 66 2e 0f 1f 84 00 00 00 00 00 |.t.]..f.....|
00000890 5d c3 0f 1f 40 00 66 2e 0f 1f 84 00 00 00 00 00 |]...@.f.....|
000008a0 48 8d 3d 89 17 20 00 48 8d 35 82 17 20 00 55 48 |H.=... .H.5... .UH|
000008b0 29 fe 48 89 e5 48 c1 fe 03 48 89 f0 48 c1 e8 3f |).H..H...H..H..?|
000008c0 48 01 c6 48 d1 fe 74 18 48 8b 05 21 17 20 00 48 |H..H..t.H..!. .H|
000008d0 85 c0 74 0c 5d ff e0 66 0f 1f 84 00 00 00 00 00 |..t.].f.....|
000008e0 5d c3 0f 1f 40 00 66 2e 0f 1f 84 00 00 00 00 00 |]...@.f.....|
000008f0 80 3d 41 17 20 00 00 75 2f 48 83 3d f7 16 20 00 |.=A. .u/H.=... .|
00000900 00 55 48 89 e5 74 0c 48 8b 3d fa 16 20 00 e8 0d |.UH..t.H.=... .|

```

- d. Sobre las páginas que no pueden ser halladas. Me parece que revisando cómo se presenta en el mapa de memoria, se puede observar que algunas de esas páginas cuentan con basura, por lo que era es mi suposición. Completamente mal estaba, pues ya revisando mejor en una referencias, hablan de que en realidad es más bien porque esa parte son secciones de memoria anónima que sirven como espacios de intercambio, por lo que no es sencillo definir las o describirlas dentro de un tipo en específico.