

Variables locales: 0x7fff685a67f4, 0x7fff685a67f8, 0x7fff685a67fc  
Variables estaticas: 0x560f42cb6c40, 0x560f42cb6c44, 0x560f42cb6c48  
Var (N.I.)estaticas: 0x560f42cb6c4c, 0x560f42cb6c50, 0x560f42cb6c54  
Variables globales: 0x560f42cb63f8, 0x560f42cb63fc, 0x560f42cb6400  
Var (N.I.)globales: 0x560f42cb6404, 0x560f42cb6408, 0x560f42cb640c  
Funciones programa: 0x560f42ca8fb4, 0x560f42ca9081, 0x560f42ca97e2  
Funciones libreria: 0x7f02264a50a0, 0x7f02264606f0, 0x7f02265b19e0

- Variables locales: Direcciones de memoria en el stack donde se almacenan estas variables locales.
- Variables estáticas: Se almacenan en la sección de datos estáticos de la memoria del programa
- Variables globales: Se almacenan en la sección de datos globales de la memoria del programa
- Funciones librería: Son direcciones en la sección de código de la biblioteca.

Shared: No tiene un lugar específico aunque suele hacerse en la RAM.

Malloc: Asigna memoria en el heap del programa.

Mmap: Asigna memoria en el espacio de direcciones de un proceso.

Recurse 4:

parametro: 4(0x7ffe502ba42c) array 0x7ffe502ba430, arr estatico 0x55eba41fb440

parametro: 3(0x7ffe502b9bfc) array 0x7ffe502b9c00, arr estatico 0x55eba41fb440

parametro: 2(0x7ffe502b93cc) array 0x7ffe502b93d0, arr estatico 0x55eba41fb440

parametro: 1(0x7ffe502b8b9c) array 0x7ffe502b8ba0, arr estatico 0x55eba41fb440

parametro: 0(0x7ffe502b836c) array 0x7ffe502b8370, arr estatico 0x55eba41fb440

Parámetro y Array se almacenan en la pila del usuario

Array Estático: Se guarda en las direcciones de datos estáticos y no cambia por su estabilidad

Partes Memoria:

1. Texto o Código: Esta es la sección de la memoria donde se almacena el código ejecutable del programa. Aquí es donde se encuentran las instrucciones que la CPU ejecuta.
2. Datos Inicializados: Esta sección de la memoria almacena las variables globales y estáticas que han sido inicializadas por el programador.
3. Datos No Inicializados / BSS: Aquí se almacenan las variables globales y estáticas que no han sido inicializadas. En muchos sistemas, esta sección de la memoria se inicializa automáticamente a cero antes de que se ejecute el programa.
4. Heap: Esta es una región de memoria que se utiliza para la asignación dinámica de memoria durante la ejecución del programa. Cuando llamas a funciones como `malloc` en C, la memoria se asigna desde el heap.
5. Stack: Esta es la región de memoria donde se almacenan las variables locales y la información de control para las funciones. Cada vez que se llama a una función, se crea un nuevo marco de pila que contiene las variables locales de la función y la información necesaria para mantener un seguimiento de la ejecución del programa.

6. Memoria Compartida: Esta es una sección especial de la memoria que puede ser compartida entre procesos. Se utiliza para la comunicación entre procesos y para evitar copias redundantes de datos.