

Activity 10.1 – Memory Management

Modify the above program by adding the function “sleep(10000)”. You may need to import the unistd.h library. Then compile the program and run it in the background. Identify the process id of this program and check the value of /proc//maps file.

Add sleep(10000) at the end of the program.

```
GNU nano 6.2 malloc.c
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
int main(){
    int *ptr;
    ptr = (int *) malloc (5 * sizeof(int));
    if(ptr != NULL){
        printf("Memory has been successfully allocated.\n");
        printf("Starting address: %p\n", ptr);
        printf("End address: %p\n", ptr+4);
        for(int i=0;i<5;i++){
            ptr[i] = i+1;
        }
        printf("The elements of the array are:\n");
        for(int i=0;i<5;i++){
            printf("%d\n", ptr[i]);
        }
    }
    sleep(10000);
}
```

1. In what part of the memory is the pointer pointed to?

```
root@---:~# Memory has been successfully allocated.
Starting address: 0x55ed96c8a2a0
End address: 0x55ed96c8a2b0
The elements of the array are:
1
2
3
4
5

root@---:~# ps
  PID TTY          TIME CMD
   10 pts/0    00:00:00 bash
  135 pts/0    00:00:00 malloc.out
  147 pts/0    00:00:00 malloc.out
  148 pts/0    00:00:00 ps

root@---:~# cat /proc/147/maps
55ed9567e000-55ed9567f000 r--p 00000000 08:10 2373 /root/malloc.out
55ed9567f000-55ed95680000 r-xp 00001000 08:10 2373 /root/malloc.out
55ed95680000-55ed95681000 r--p 00002000 08:10 2373 /root/malloc.out
55ed95681000-55ed95682000 r--p 00002000 08:10 2373 /root/malloc.out
55ed95682000-55ed95683000 rwxp 00003000 08:10 2373 /root/malloc.out
55ed96c8a000-55ed96cab000 rw-p 00000000 00:00 0 [heap]
7f1de935b000-7f1de935b000 rw-p 00000000 00:00 0
7f1de935b000-7f1de9383000 r--p 00000000 08:10 12498 /usr/lib/x86_64-l
7f1de9383000-7f1de9518000 r-xp 00028000 08:10 12498 /usr/lib/x86_64-l
7f1de9518000-7f1de9570000 r--p 001bd000 08:10 12498 /usr/lib/x86_64-l
7f1de9570000-7f1de9574000 r--p 00214000 08:10 12498 /usr/lib/x86_64-l
7f1de9574000-7f1de9576000 rw-p 00218000 08:10 12498 /usr/lib/x86_64-l
7f1de9576000-7f1de9583000 rw-p 00000000 00:00 0
```

Answer: The pointer is pointed into the [heap] section of the memory

2. What does it mean in terms of memory allocation?

Answer: In terms of memory allocation, if we put our running program in the background, we could observe that by using the method above, it will put both of our starting address (0x55ed96c8a2a0) and end address (0x55ed96c8a2b0) at address 55ed96c8a exactly on the [heap] which depicted on the image above.