

REPORT

Part 2

mymodule.c is a module that will be inserted to the kernel in order to display memory management of a specific process.

After make, by typing “sudo insmod ./mymodule.ko processid=K” the module is inserted. K is the process id whose memory management will be displayed.

Firstly, it will output the name and id of the process in the form of name_of_process[pid_of_process].

Secondly, start, end, and size of the code's, data's, stack's, heap's, main arguments', environment variables' virtual addresses, number of frames used by the process and its size, total virtual memory used by the process. Addresses are printed as hexadecimal, sizes are printed as decimal numbers.

Thirdly, top level page's entry number and entry.

In order to print virtual addresses of memory blocks of the process, attributes in mm_struct belonging to the input process are used. Only for stack, its virtual memory address's start and end addresses are used. For top page table entry, pgd_t pgd attribute of mm_struct is used. 512 entries of pgd are parsed and then the valid ones are printed.

An example output of this module will be given below.

```
cs342@cs342vm: ~/Desktop/project3
cs342@cs342vm:~/Desktop/project3$ dmesg
[ 1906.950248] PID IS app[4772]
[ 1906.950257] Size of virtual memory = 4352 KB
                Number of frames = 183
                Size of used frames = 732 KB
                Code Segment start address= 0x400000, end address= 0x400d4c, size = 3404 ,B
                Data Segment start address= 0x600e10, end address= 0x601068, size = 600 B
                Heap Segment start address= 0x2178000, end address= 0x2199000, size = 135168 B
                Stack Segment start address= 0x7ffef740c000, end address= 0x7ffef742d000, size = 135168 B
                Main Arguments start address= 0x7ffef742c214, end address= 0x7ffef742c21a, size = 6 B
                Environment variables start address= 0x7ffef742c21a, end address= 0x7ffef742cff2, size = 3544 B
[ 1906.950258] 0th entry = 0x8000000077b34067
[ 1906.950260] 255th entry = 0x8000000077b88067
[ 1906.950261] 314th entry = 0x76f2a067
[ 1906.950262] 387th entry = 0x7d138067
[ 1906.950263] 451th entry = 0x71ac0067
[ 1906.950264] 499th entry = 0x7f73a067
[ 1906.950300] 508th entry = 0x7ff2f067
[ 1906.950300] 510th entry = 0x76cde067
[ 1906.950301] 511th entry = 0x7680e067
cs342@cs342vm:~/Desktop/project3$
```

Part 3

app.c is designed in a way that it asks the user to choose an option repeatedly, until he chooses to exit. These are:

- Option 0: Exit
- Option 1: Call a recursive function, in other terms, use the stack
- Option 2: Allocate memory dynamically, in other terms, use the heap

During these steps, we look the status of the stack and heap by checking with the command “cat /proc/pid/maps”. We get the process id of app.c by getpid() which is at the beginning of the main fuction. Also, we check virtual memory size and number of frames by the command “ps aux” at each step.

Stack

main function calls a recursive function f which has only one integer parameter, x . At the beginning, this value is 1. $f(x)$ calls $f(x+1)$ repeatedly. Whenever $x \bmod 10000$ is equal to 0, it asks to continue or not. Instead of putting `sleep(sec)`, we checked the stack in this way. At the beginning, size of the stack is 135168 B where its start address is 0x7ffd59d1a000 and its end address is 0x7ffd59d3b000, and VM size is 4352 KB, and number of frames is equal to 680. We observed that when $x = 100000$, its size becomes 4812800 B and its start address changes to 0x7ffd598a4000, and VM size is 8920 KB, and number of frames is equal to 5824.

The image shows a Kali Linux desktop environment. On the left side, there is a vertical dock containing icons for various applications, including a web browser, file manager, and terminal. The main area of the screen is occupied by two terminal windows. The left terminal window has a title bar that reads 'metehan@metehan-N56VZ: ~/Masaüstü/CS342/Project3'. It shows the execution of the command 'insmod ./nynodule3.ko' followed by 'dmesg -c'. The output displays kernel memory statistics, including virtual memory size (4352 KB), number of frames (RSS) (680), and segment start addresses. It also shows the main arguments and environment variables for the module. The right terminal window has a title bar that reads 'metehan@metehan-N56VZ: ~/Masaüstü/CS342/Project3'. It shows the execution of the command './app'. The output displays a menu with options to print 0 to exit, print 0 to return from func, print 1 to continue to func, and print 2 to use heap. It also shows recursive steps and environment variables. The desktop background is black, and the overall layout is typical of a Linux desktop environment.

Figure 3a: initial status of the stack

```
Ucbrim metehan@metehan-NS6VZ:~/MasauStu/CS342/Project3
3919.103654 48th entry = 0x226c2067
3919.103654 508th entry = 0x22f1be07
3919.103655 510th entry = 0x108df067
3919.103655 511th entry = 0x1040e067
3919.484772
metehan@metehan-NS6VZ:~/MasauStu/CS342/Project3$ sudo lsmmod ./nynodule3.ko processid=4984
metehan@metehan-NS6VZ:~/MasauStu/CS342/Project3$ sudo rmmod ./nynodule3.ko
metehan@metehan-NS6VZ:~/MasauStu/CS342/Project3$ sudo dmesg -c
3959.448381 PID 15: app(4984)
3959.448391 Virtual Memory size = 8920 KB
Number of frames(RSS) = 5824
Code Segment start address= 0x400000, end address= 0x40004c, size = 3404 B
Data Segment start address= 0x600e10, end address= 0x601068, size = 600 B
Heap Segment start address= 0xc6e000, end address= 0xc8f000, size = 135168 B
Stack Segment start address= 0x7ff598a4000, end address= 0x7ff59d3b000, size = 4812 B
Main Arguments start address= 0x7ff59d3a28e, end address= 0x7ff59d3a294, size = 6 B
Environment Variables start address= 0x7ff59d3a294, end address= 0x7ff59d3aff2, size = 3422 B
3959.448393 0th entry = 0x80000001ac585067
3959.448395 254th entry = 0x800000017bd0067
3959.448396 255th entry = 0x80000001c64e067
3959.448397 312th entry = 0x10b2c067
3959.448399 371th entry = 0x226537067
3959.448400 435th entry = 0x21ba78067
3959.448401 486th entry = 0x226c2067
3959.448402 508th entry = 0x22f1be07
3959.448404 510th entry = 0x108df067
3959.448405 511th entry = 0x1040e067
3951.820651
metehan@metehan-NS6VZ:~/MasauStu/CS342/Project3$ cat /proc/4984/maps
00400000-00401000 r-xp 00000000 08:07 556478 /home/metehan/Masa0stu/CS342
/Project3/app
00600000-00602000 rw-p 00000000 08:07 556478 /home/metehan/Masa0stu/CS342
/Project3/app
008c0000-008ee000 rw-p 00000000 00:00 0 /home/metehan/Masa0stu/CS342
/Project3/app
008cd000-008ee000 rw-p 00000000 00:00 0 [heap]
7f73db5f000-7f73dd1f000 r-xp 00000000 08:07 409806 /lib/x86_64-linux-gnu/libc-2
.23.so
7f73dd1f000-7f73dd1f000 --p 001c0000 08:07 409806 /lib/x86_64-linux-gnu/libc-2
.23.so
7f73dd1f000-7f73dd23000 r-p 001c0000 08:07 409806 /lib/x86_64-linux-gnu/libc-2
.23.so
7f73dd23000-7f73dd25000 rw-p 00000000 08:07 409806 /lib/x86_64-linux-gnu/libc-2
.23.so
7f73dd25000-7f73dd29000 rw-p 00000000 00:00 0 [vdso]
7f73dd29000-7f73dd4f000 r-xp 00000000 08:07 409804 /lib/x86_64-linux-gnu/ld-2.2
3.so
7f73e135000-7f73e138000 rw-p 00000000 00:00 0 [vsyscall]
7f73e14e000-7f73e14f000 r-p 00025000 08:07 409804 /lib/x86_64-linux-gnu/ld-2.2
3.so
7f73e14f000-7f73e150000 rw-p 00026000 08:07 409804 /lib/x86_64-linux-gnu/ld-2.2
3.so
7f73e150000-7f73e151000 rw-p 00000000 00:00 0 [stack]
7f73e151000-7f73e152000 r-p 00000000 00:00 0 [vdso]
7f73e152000-7f73e153000 r-xp 00000000 00:00 0 [vsyscall]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0
metehan@metehan-NS6VZ:~/MasauStu/CS342/Project3$
```

Figure 3b: final status of the stack (with cat /proc/pid/maps command)

```
Ucbrim metehan@metehan-NS6VZ:~/MasauStu/CS342/Project3
metehan 2367 0.0 0.4 662446 36760 ? Ssl 19:02 0:01 goany
metehan 2383 0.2 0.4 664880 38468 ? Ssl 19:04 0:00 /usr/lib/gnome-terminal/gnome-termi
metehan 2389 0.0 0.0 24076 5248 pts/17 Ss 19:04 0:00 bash
metehan 2407 0.0 0.0 23944 5128 pts/18 Ss 19:04 0:00 bash
metehan 2411 5.5 3.1 1550404 75408 ? Ssl 19:04 3:10 /opt/google/chrome/chrome
metehan 2437 0.0 0.0 8852 768 ? S 19:04 0:00 cat
metehan 2438 0.0 0.0 8852 820 ? S 19:04 0:00 cat
metehan 2441 0.0 0.0 62166 5684 ? S 19:04 0:00 /opt/google/chrome/chrome --type=zyg
metehan 2442 0.0 0.0 25720 4096 ? S 19:04 0:00 /opt/google/chrome/chrome --type=ren
metehan 2445 0.0 0.1 421268 14028 ? S 19:04 0:00 /opt/google/chrome/chrome --type=zyg
metehan 2498 3.1 2.0 607428 166472 ? S 19:04 2:04 /opt/google/chrome/chrome --type=zyg
metehan 2502 0.0 0.1 450948 14168 ? S 19:04 0:00 /opt/google/chrome/chrome --type=br
metehan 2511 0.0 0.3 447784 27316 ? S 19:04 0:00 /usr/lib/x86_64-linux-gnu/notify-0sd
metehan 2574 0.5 2.9 1632552 248140 ? S 19:04 0:23 /opt/google/chrome/chrome --type=ren
metehan 2605 0.0 0.2 553416 19696 ? S 19:04 0:01 zeitgeist-database
metehan 2615 0.0 0.0 4504 756 ? S 19:04 0:00 /bin/sh -c /usr/lib/x86_64-linux-gnu
metehan 2619 0.0 0.1 421384 9348 ? S 19:04 0:00 /usr/bin/zeitgeist-daemon
metehan 2641 0.0 0.1 319328 15436 ? S 19:04 0:00 /usr/lib/x86_64-linux-gnu/zeitgeist-
metehan 2691 1.0 5.0 1493292 408136 ? S 19:04 0:40 /opt/google/chrome/chrome --type=ren
metehan 2709 1.5 3.3 1345460 278852 ? S 19:04 1:00 /opt/google/chrome/chrome --type=ren
metehan 2729 0.4 2.1 2851120 172696 ? S 19:04 0:18 /opt/google/chrome/chrome --type=ren
metehan 2782 0.0 0.0 8817668 7640 ? S 19:04 0:00 /opt/google/chrome/chrome --type=ren
metehan 2876 0.0 0.2 524988 23784 ? S 19:04 0:00 update-notifier
metehan 2919 0.0 0.1 363520 8412 ? S 19:05 0:00 /usr/lib/gvfs/gvfsd-network --spawne
metehan 2967 0.0 0.0 363304 6876 ? S 19:05 0:00 /usr/lib/gvfs/gvfsd-dnssd --spawne
metehan 3032 0.0 0.1 442860 8672 ? S 19:05 0:00 /usr/lib/x86_64-linux-gnu/deja-dup/d
metehan 3382 0.0 0.0 135356 5820 ? S 19:13 0:00 /usr/lib/libreoffice/program/soffice
metehan 3400 0.5 2.7 2227216 219032 ? S 19:13 2:00 /usr/lib/libreoffice/program/soffice
root 3435 0.0 0.0 0 0 ? S 19:13 0:00 [kworker/4:0]
metehan 4018 11.6 6.1 1620816 497856 ? S 19:18 5:56 /opt/google/chrome/chrome --type=ren
root 4316 0.0 0.0 0 0 ? S 19:26 0:00 [kworker/1:0]
root 4317 0.0 0.0 0 0 ? S 19:26 0:00 [kworker/0:12]
root 4346 0.0 0.0 13952 2496 ? S 19:26 0:00 /sbin/mount.ntfs /dev/sdb2 /media/me
root 4425 0.0 0.0 0 0 ? S 19:27 0:00 [kworker/5:0]
root 4785 0.0 0.0 0 0 ? S 19:32 0:01 [kworker/u6:4]
root 4728 0.0 0.0 0 0 ? S 19:33 0:00 [kworker/2:1]
root 4937 0.0 0.0 0 0 ? S 19:37 0:00 [kworker/7:2]
metehan 4934 0.0 0.0 8920 5824 pts/18 Ss 19:30 0:00 /app
root 5019 0.0 0.0 0 0 ? S 19:40 0:00 [kworker/3:1]
root 5157 0.0 0.0 0 0 ? S 19:40 0:00 [kworker/6:1]
root 5201 0.0 0.0 0 0 ? S 19:40 0:00 [kworker/u6:0]
metehan 5265 0.0 2.2 646520 17412 ? S 19:50 0:00 /usr/lib/x86_64-linux-gnu/unity-scop
metehan 5283 0.0 0.3 651288 24404 ? S 19:50 0:00 /usr/bin/unity-scope-loader applicat
metehan 5285 0.0 0.1 681544 13676 ? S 19:50 0:00 /usr/lib/x86_64-linux-gnu/unity-lens
root 5393 0.0 0.0 0 0 ? S 19:51 0:00 [kworker/4:2]
root 5396 0.0 0.0 0 0 ? S 19:51 0:00 [kworker/3:0]
root 5452 0.0 0.0 0 0 ? S 19:52 0:00 [kworker/1:2]
root 5473 0.0 0.0 0 0 ? S 19:53 0:00 [kworker/5:2]
root 5500 0.0 0.0 0 0 ? S 19:54 0:00 [kworker/0:0]
root 5527 0.0 0.0 0 0 ? S 19:55 0:00 [kworker/7:1]
root 5566 0.0 0.0 0 0 ? S 19:56 0:00 [kworker/u6:1]
metehan 5592 0.6 0.7 716424 63924 ? S 19:57 0:04 /opt/home/metehan/Masa0stu/ss6.png
root 5684 0.0 0.0 0 0 ? S 19:59 0:00 [kworker/2:2]
root 5911 0.0 0.0 0 0 ? S 20:04 0:00 [kworker/u6:2]
root 5944 0.0 0.0 0 0 ? S 20:05 0:00 [kworker/4:1]
root 6125 0.0 0.0 0 0 ? S 20:09 0:00 [kworker/7:0]
root 6128 0.0 0.0 0 0 ? S 20:09 0:00 [kworker/5:1]
metehan 6135 0.0 0.0 38780 3432 pts/17 R 20:09 0:00 ps aux
metehan@metehan-NS6VZ:~/MasauStu/CS342/Project3$
```

Figure 3c: final status of the stack (with ps aux command)

Heap

In main function, there is a loop that allocates 10000 integers at each step. After it allocates memory dynamically, asks for user's choice, to continue or not. At the beginning, size of the heap is 135168 B where its start address is 0xc6e000 and its end address is 0xc8f000, and VM size is 4352 KB, and number of frames is equal to 161. After 100000 integers are allocated, heap's end address

becomes 0xcd000, and its size becomes 454656 B, and VM size becomes 4664 KB, and number of frames is equal to 161 which did not change. What we observed that, whenever stack size increases, VM size increases too.

```
metehan@metehan-N56VZ: ~/MasauStu/CS342/Project3
metehan@metehan-N56VZ:~/MasauStu/CS342/Project3$ sudo insmod ./nynodule.ko processid=7492
metehan@metehan-N56VZ:~/MasauStu/CS342/Project3$ sudo rmmod ./nynodule.ko
metehan@metehan-N56VZ:~/MasauStu/CS342/Project3$ sudo dmesg -c
[ 5680.401017] PID 15 app[7492]
[ 5680.401027] Size of virtual memory = 4352 KB
[ 5680.401027] Number of frames = 161
[ 5680.401027] Size of used frames = 644
[ 5680.401024] Code Segment start address= 0x400000, end address= 0x40004c, size = 3404 B
[ 5680.401024] Data Segment start address= 0x600e10, end address= 0x601068, size = 600 B
[ 5680.401026] Heap Segment start address= 0xc60000, end address= 0xc6f000, size = 135168 B
[ 5680.401026] Stack Segment start address= 0x7ffe54a72000, end address= 0x7ffe54a48000, size = 1351 B
[ 5680.401027] Main Arguments start address= 0x7ffe54a4728e, end address= 0x7ffe54a47294, size = 6 B
[ 5680.401027] Environment variables start address= 0x7ffe54a47294, end address= 0x7ffe54a47ff2, size = 3422 B
[ 5680.401023] 0th entry = 0x8000000b411067
[ 5680.401024] 254th entry = 0x80000001a70d4067
[ 5680.401024] 255th entry = 0x80000001ac6a8067
[ 5680.401025] 312th entry = 0x10b2c067
[ 5680.401025] 371th entry = 0x226537067
[ 5680.401026] 435th entry = 0x21b78067
[ 5680.401026] 486th entry = 0x22e9c2067
[ 5680.401027] 508th entry = 0x22f1be067
[ 5680.401027] 510th entry = 0x108df067
[ 5680.401028] 511th entry = 0x1040e067
[ 5680.763211] [redacted]
metehan@metehan-N56VZ:~/MasauStu/CS342/Project3$ sudo insmod ./nynodule.ko processid=7492
metehan@metehan-N56VZ:~/MasauStu/CS342/Project3$ cat /proc/7492/maps
00400000-00401000 r-xp 00000000 08:07 576075 /home/metehan/MasauStu/CS342
/Project3/app
00600000-00601000 r--p 00000000 08:07 576075 /home/metehan/MasauStu/CS342
/Project3/app
00601000-00602000 rw-p 00001000 08:07 576075 /home/metehan/MasauStu/CS342
/Project3/app
00ce0000-00cd0000 rw-p 00000000 00:00 0 [heap]
7f3edd20000-7f3edd3c000 r-xp 00000000 08:07 409806 /lib/x86_64-linux-gnu/libc-2
.23.so
7f3edd3c000-7f3edd5c000 ---p 001c0000 08:07 409806 /lib/x86_64-linux-gnu/libc-2
.23.so
7f3edd5c000-7f3edd5cc000 r-p 001c0000 08:07 409806 /lib/x86_64-linux-gnu/libc-2
.23.so
7f3edd5cc000-7f3edd5ce000 rw-p 001c4000 08:07 409806 /lib/x86_64-linux-gnu/libc-2
.23.so
7f3edd5ce000-7f3edd5d2000 rw-p 00000000 00:00 0 /lib/x86_64-linux-gnu/ld-2.2
7f3edd5d2000-7f3edd5f8000 r-xp 00000000 08:07 409804 /lib/x86_64-linux-gnu/ld-2.2
3.so
7f3edd7de000-7f3edd7e1000 rw-p 00000000 00:00 0 /lib/x86_64-linux-gnu/ld-2.2
3.so
7f3edd7f000-7f3edd7f8000 r-p 00025000 08:07 409804 /lib/x86_64-linux-gnu/ld-2.2
3.so
7f3edd7f8000-7f3edd7f9000 rw-p 00020000 08:07 409804 /lib/x86_64-linux-gnu/ld-2.2
3.so
7f3edd7f9000-7f3edd7fa000 rw-p 00000000 00:00 0
7ffe54a72000-7ffe54a8000 rw-p 00000000 00:00 0
7ffe54ac2000-7ffe54ac5000 r-p 00000000 00:00 0
7ffe54ac5000-7ffe54ac7000 r-xp 00000000 00:00 0
7ffe54ac7000-7ffe54ac7000 r-xp 00000000 00:00 0
ffffffffff000000-ffffffffff001000 r-xp 00000000 00:00 0
metehan@metehan-N56VZ:~/MasauStu/CS342/Project3$
```

Figure 3d: initial status of the heap

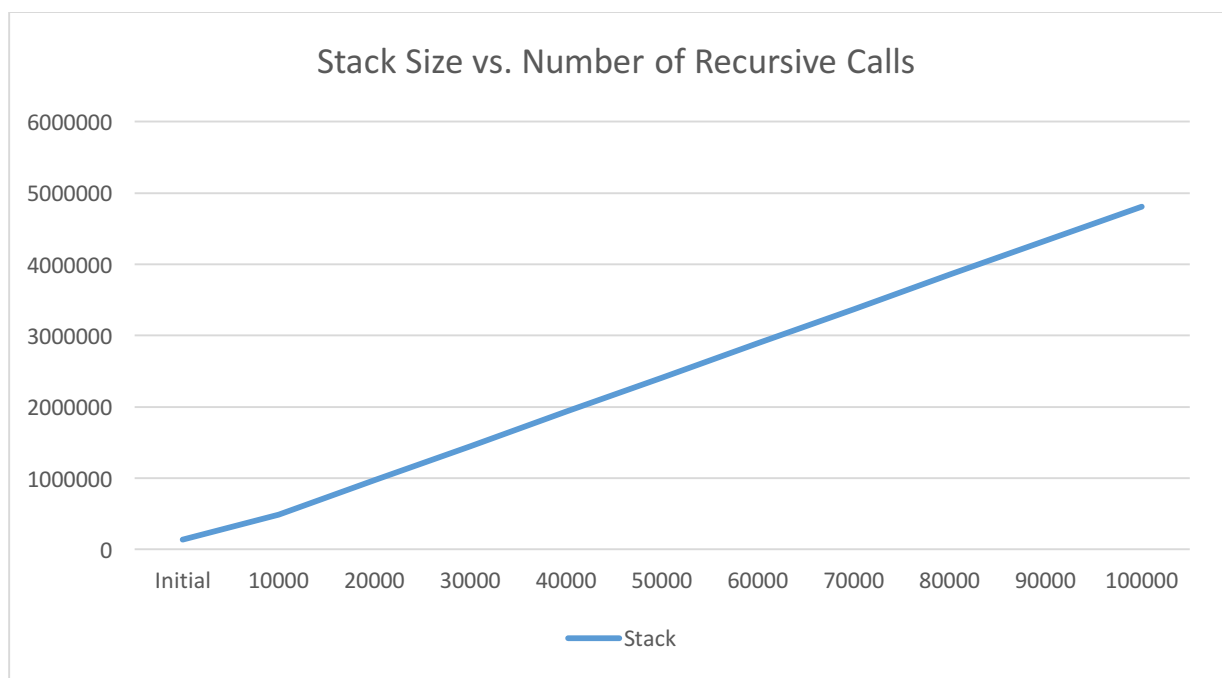
```
metehan@metehan-N56VZ:~/MasauStu/CS342/Project3
metehan@metehan-N56VZ:~/MasauStu/CS342/Project3$ sudo insmod ./nynodule.ko processid=7492
metehan@metehan-N56VZ:~/MasauStu/CS342/Project3$ sudo rmmod ./nynodule.ko
metehan@metehan-N56VZ:~/MasauStu/CS342/Project3$ sudo dmesg -c
[ 7598.808016] PID 15 app[7492]
[ 7598.808016] Size of virtual memory = 4664 KB
[ 7598.808016] Number of frames = 161
[ 7598.808016] Size of used frames = 644
[ 7598.808016] Code Segment start address= 0x400000, end address= 0x40004c, size = 3404 B
[ 7598.808016] Data Segment start address= 0x600e10, end address= 0x601068, size = 600 B
[ 7598.808016] Heap Segment start address= 0xc60000, end address= 0xc6f000, size = 135168 B
[ 7598.808016] Stack Segment start address= 0x7ffe54a72000, end address= 0x7ffe54a48000, size = 1351 B
[ 7598.808016] Main Arguments start address= 0x7ffe54a4728e, end address= 0x7ffe54a47294, size = 6 B
[ 7598.808016] Environment variables start address= 0x7ffe54a47294, end address= 0x7ffe54a47ff2, size = 3422 B
[ 7598.808016] 0th entry = 0x8000000b411067
[ 7598.808017] 254th entry = 0x80000001a70d4067
[ 7598.808018] 255th entry = 0x80000001ac6a8067
[ 7598.808020] 312th entry = 0x10b2c067
[ 7598.808021] 371th entry = 0x226537067
[ 7598.808022] 435th entry = 0x21b78067
[ 7598.808023] 486th entry = 0x22e9c2067
[ 7598.808024] 508th entry = 0x22f1be067
[ 7598.808025] 510th entry = 0x108df067
[ 7598.808025] 511th entry = 0x1040e067
[ 7598.953079] [redacted]
metehan@metehan-N56VZ:~/MasauStu/CS342/Project3$ cat /proc/7492/maps
00400000-00401000 r-xp 00000000 08:07 576075 /home/metehan/MasauStu/CS342
/Project3/app
00600000-00601000 r--p 00000000 08:07 576075 /home/metehan/MasauStu/CS342
/Project3/app
00601000-00602000 rw-p 00001000 08:07 576075 /home/metehan/MasauStu/CS342
/Project3/app
00ce0000-00cd0000 rw-p 00000000 00:00 0 [heap]
7f3edd20000-7f3edd3c000 r-xp 00000000 08:07 409806 /lib/x86_64-linux-gnu/libc-2
.23.so
7f3edd3c000-7f3edd5c000 ---p 001c0000 08:07 409806 /lib/x86_64-linux-gnu/libc-2
.23.so
7f3edd5c000-7f3edd5cc000 r-p 001c0000 08:07 409806 /lib/x86_64-linux-gnu/libc-2
.23.so
7f3edd5cc000-7f3edd5ce000 rw-p 001c4000 08:07 409806 /lib/x86_64-linux-gnu/libc-2
.23.so
7f3edd5ce000-7f3edd5d2000 rw-p 00000000 00:00 0 /lib/x86_64-linux-gnu/ld-2.2
7f3edd5d2000-7f3edd5f8000 r-xp 00000000 08:07 409804 /lib/x86_64-linux-gnu/ld-2.2
3.so
7f3edd7de000-7f3edd7e1000 rw-p 00000000 00:00 0 /lib/x86_64-linux-gnu/ld-2.2
3.so
7f3edd7f000-7f3edd7f8000 r-p 00025000 08:07 409804 /lib/x86_64-linux-gnu/ld-2.2
3.so
7f3edd7f8000-7f3edd7f9000 rw-p 00020000 08:07 409804 /lib/x86_64-linux-gnu/ld-2.2
3.so
7f3edd7f9000-7f3edd7fa000 rw-p 00000000 00:00 0
7ffe54a72000-7ffe54a8000 rw-p 00000000 00:00 0
7ffe54ac2000-7ffe54ac5000 r-p 00000000 00:00 0
7ffe54ac5000-7ffe54ac7000 r-xp 00000000 00:00 0
7ffe54ac7000-7ffe54ac7000 r-xp 00000000 00:00 0
ffffffffff000000-ffffffffff001000 r-xp 00000000 00:00 0
metehan@metehan-N56VZ:~/MasauStu/CS342/Project3$
```

Figure 3e: final status of the heap

Plots and Tables

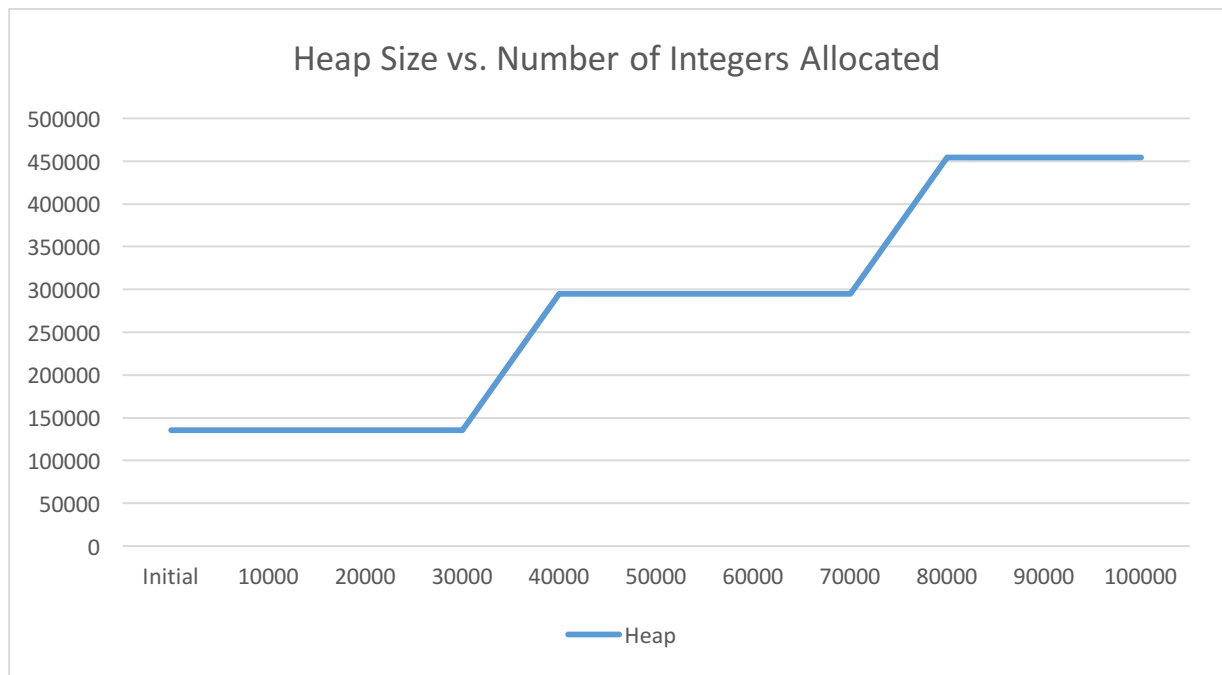
# of recursive calls (1 int for each)	Stack size (B)
Initial	135168
10000	491520
20000	970752
30000	1454080
40000	1933312
50000	2412544
60000	2891776
70000	3371008
80000	3854336
90000	4333568
100000	4812800

Table 3a: Stack



# of integers allocated	Heap size (B)
Initial	135168
10000	135168
20000	135168
30000	135168
40000	294912
50000	294912
60000	294912
70000	294912
80000	454656
90000	454656
100000	454656

Table 3b: Heap



Heap space with 135168 B was sufficient to store up to 30000 integers, similarly 294912 B was sufficient to store from 30000 to 70000 integers. Therefore, when we want to allocate 40000 and 80000 integers on the heap, its size grows immediately.

Stack grows much faster than heap because when we call a function, it will allocate more space than the same number of integers that are allocated in the heap. This is because function wants to allocate space for its parameters, code, and some other data in the stack.