CS 342

Operating Systems

Project 3

Mian Usman Naeem Kakakhel

21701015

Sec: 01

04/05/2020

NOTE: Full OBJDUMPs and all PMAPs are given at the end of the report. Necessary parts of the PMAPs and OBJDUMPs are included in the explanations as required.

List of contents of all modules:

Module1.c :

- Function main(): runs the whole program.
- Function step1(): runs the step 1 in the project.
- Function step2(): runs the step 2 in the project.
- Function step3(): runs the step 3 in the project.
- Function step4(): runs the step 4 in the project.
- Function step5(): runs the step 5 in the project.
- Function step6(): runs the step 6 in the project.
- char* region: it is the shared memory given by mmap in step 4 which will be pointing to 2MB space later on when step 4 is run.

Module2.c :

- long long initialized[16]: which is initialized to some values and since long long is 8 bytes, total memory utilized by this is 128 bytes.
- long long un_initialized[2000]: which is not initialized and the total memory utilized by this is 2 KB.
- int init_const: which is used for seeing where consts lie in VM.
- long long* un_init_ptr1: which will be used to point to 2MB of space given by malloc() in step 2.
- long long* un_init_ptr2: which will be used to point to 2MB of space given by malloc() in step 2.

Module3.c :

- Function foo(): used to run the recursive function required in step 3.

Step1:

Output to step 1:

```
initialized address: 0x6020a0
un_initialized address: 0x602160
un_init_ptr1 address: 0x602140
un_init_ptr2 address: 0x602148
const init_const: 0x401078
main_func: 0x400857
step1_func: 0x4009f1
step2_func: 0x400b72
step3_func: 0x400bd1
```

```
step4_func: 0x400be7
step5_func: 0x400c9c
step6_func: 0x400ccb
foo_func: 0x400d3d
printf_func: 0x7fd64324ae80
```

The required PMAP here:

```
0000000000400000      8K r-x-- app
0000000000601000      4K r---- app
0000000000602000      4K rw--- app
```

We can see in the virtual memory of app that the text section starts from 400770 and from our output, all functions including main, foo and all step functions are from this section except printf() because it is from the libc section. Also this is the text section as we can see that this has r-x permissions so code can be read and executed. After that, all the uninitialized and initialized are in the data section starting from 602000. This data section has permissions rw so that variables can be written and read during execution. The constant is also put in the 400000 section as the read only section starts from 400e20 and the constant address is 401078. The sections can be seen from objdump -h app. Required section is as below:

```
13 .text         000006a2  0000000000400770  0000000000400770  00000770  2**4
15 .rodata       00000283  0000000000400e20  0000000000400e20  00000e20  2**3
```

According to the Objdump:

```
00000000006020a0 g     O .data     0000000000000080       initialized
0000000000602160 g     O .bss      0000000000003e80       un_initialized
0000000000602140 g     O .bss      0000000000000008       un_init_ptr1
0000000000602148 g     O .bss      0000000000000008       un_init_ptr2
```

The addresses of the variables are in their correct sections and we can see that initialized data is kept in the data section but the uninitialized variables are kept in the bss section.

Also as shown before, the constants are kept in the readonly data section,

```
0000000000401078 g     O .rodata   0000000000000004       init_const
```

And the functions are all kept in the text section,

```
0000000000400857 g     F .text     000000000000019a       main
0000000000400d3d g     F .text     0000000000000055       foo
0000000000400c9c g     F .text     000000000000002f       step5
0000000000400bd1 g     F .text     0000000000000016       step3
00000000004009f1 g     F .text     0000000000000181       step1
0000000000400be7 g     F .text     00000000000000b5       step4
```

```
0000000000400b72 g     F .text          000000000000005f              step2
0000000000400ccb g     F .text          0000000000000072              step6
```

Thus it can be concluded that any variables, constants and code that I have for this part can be found in either .data, .bss, .rodata or .text.

Basically, the modules are called as relocatable object modules because they cannot be executed even if we try to execute them using ./module.o. Comparing the objdumps of module1.o, module2.o and app, we can see some reasons why these modules cannot be executed and are called relocatable object modules even though they are in the ELF format.

Firstly let's compare the main function in the module1.o and app:

```
0000000000400857 g     F .text          000000000000019a              main
0000000000000000 g     F .text          000000000000019a              main
```

The main of module1 has an address of 000000 while the main of app has address of 400857. This is the case with the rest of the functions and the variables in the modules and app:

module1:

```
000000000000019a g     F .text          0000000000000181 step1
000000000000031b g     F .text          000000000000005f step2
000000000000037a g     F .text          0000000000000016 step3
0000000000000390 g     F .text          00000000000000b5 step4
0000000000000445 g     F .text          000000000000002f step5
0000000000000474 g     F .text          0000000000000072 step6
```

App:

```
0000000000400c9c g     F .text          000000000000002f              step5
0000000000400bd1 g     F .text          0000000000000016              step3
00000000004009f1 g     F .text          0000000000000181              step1
0000000000400be7 g     F .text          00000000000000b5              step4
0000000000400b72 g     F .text          000000000000005f              step2
0000000000400ccb g     F .text          0000000000000072              step6
```

Module1:

```
0000000000000000 g     O .data          0000000000000080 initialized
0000000000003e80       O *COM*          0000000000000020 un_initialized
0000000000000008       O *COM*          0000000000000008 un_init_ptr1
0000000000000008       O *COM*          0000000000000008 un_init_ptr2
0000000000000000 g     O .rodata        0000000000000004 init_const
```

App:

```
0000000000602140 g     O .bss           0000000000000008              un_init_ptr1
00000000006020a0 g     O .data          0000000000000080              initialized
0000000000602148 g     O .bss           0000000000000008              un_init_ptr2
```

```
0000000000602160 g     O .bss        0000000000003e80            un_initialized
0000000000401078 g     O .rodata     0000000000000004            init_const
```

We can see that their addresses are obviously different and module2 does not even have a bss section. As a matter of fact, the module files are missing a lot of sections, which can be seen in the full output of sections of objdumps of modules and app, due to which we can conclude that these modules cannot be executed and thus are called relocatable object modules.

In the app objdump, the references of variables and constants are as below:

```
0000000000000004            init_const
0000000000003e80            un_initialized
0000000000000080            initialized
0000000000000008            un_init_ptr1
0000000000000008            un_init_ptr2
```

In the module2 objdump, the references of variables are as follows:

```
0000000000000080    initialized
0000000000000020    un_initialized
0000000000000008    un_init_ptr1
0000000000000008    un_init_ptr2
0000000000000004    init_const
```

In the app objdump, the references of functions are as below:

```
000000000000019a            main
0000000000000055            foo
000000000000002f            step5
0000000000000016            step3
0000000000000181            step1
00000000000000b5            step4
000000000000005f            step2
0000000000000072            step6
```

In the module 1 and 3, the references of functions are as below:

```
0000000000000181    step1
000000000000005f    step2
0000000000000016    step3
00000000000000b5    step4
000000000000002f    step5
0000000000000072    step6
000000000000019a    main
0000000000000055    foo
```

We can clearly see from these outputs that the references for the functions and variables in the modules are the same as the references given by the objdump of the app because the sections are starting from different addresses but inside the different sections, the offset of these variables and functions are the same thus the references in the app and modules are the same.

Printf from objdump of app:

```
 400760:        ff 25 82 18 20 00       jmpq   *0x201882(%rip)        # 601fe8
<printf@GLIBC_2.2.5>
```

And while running, printf has address,

```
printf_func: 0x7fd64324ae80
```

Thus we can see that the address of printf while running is in the libc section and then the reference given to it is 400760.

From inspection of the files in the /proc/pid directory it can be seen that these files are somewhat related to the Virtual memory:

1. Limits
2. Maps
3. Pagemap
4. Smaps
5. Stack
6. Stat
7. Statm
8. status

Step 2:

Output of step2 is:

```
address at un_init_ptr1 : 0x7fd642ffd010
address at un_init_ptr2 : 0x7fd642e14010
```

pmap step2 has:

```
00007fd642e14000   3912K rw---   [ anon ]
```

We can see that the addresses to which these ptrs are pointing are in this section which is given above with an area allocation of 3912K which is totally according to the 4MB data that I had requested from malloc().

Pmap -x pid has:

```
Address           Kbytes     RSS   Dirty Mode  Mapping
00007fd642e14000    3912       8       8 rw---   [ anon ]
```

We can see that the resident set size (RSS) is 8KB which is 2 pages even though virtual memory is 4MB which is assigned to us.

Step 3:

Called the foo function which calls itself recursively 12500 times. And inside foo I declare a long long, thus 8 * 12500 is 100 KB. And then I print the first and last long long created on the stack.

Output of step3:

```
address of var inside foo: 0x7ffd697bf1fc
address of var inside foo: 0x7ffd6972ca6c
```

Pmap output:

```
00007ffd6972c000     596     596     596 rw---   [ stack ]
```

We can see that the addresses of the first and end long long on the recursive calls are greater than the stack address in pmap and also the stack has 596 KB of space even though I got 100 KB of local variables declared. This is because function recursion also has its own overhead plus I have also declared int in the parameter.

Step 4:

Output of step 4:

```
address mapped mem : 0x7fd642c2b000
```

Pmap output:

```
00007fd642c2b000    1956       0       0 rw--- shm.txt
```

We can see that pmap has created a new section by the name of the shared memory file and has given it a size of 1MB in virtual but 0KB in physical memory. The start address printed is the same as the section address in pmap.

Step 5:

I wrote 6000 characters which are 6 KBs. Since the page size is 4KB, the physical memory assigned to the shared memory is 8KB as shown below:

```
00007fd642c2b000    1956       8       8 rw--- shm.txt
```

From the output of PMAP(given at the end) we can conclude the following sizes of the sections in the VM:

Data: 4 + 3912 / 4 = 979 pages

Code: 8 / 4 = 2 pages

Shared memory: 1956 / 4 = 489 pages

Stack: 596 / 4 = 149 pages

The following is the explanation of the elements of pmap:

```
0000000000400000      8K r-x-- app
0000000000601000      4K r---- app
0000000000602000      4K rw--- app
```

These are the text, data and read only data of the app.

```
0000000000603000     12K rw---   [ anon ]
0000000002508000    132K rw---   [ anon ]
```

These were the same in step 1 so do not know what they do.

```
00007fd642c2b000   1956K rw--- shm.txt
```

This is the shared memory section.

```
00007fd642e14000   3912K rw---   [ anon ]
```

This is the heap section.

```
00007fd6431e6000   1948K r-x-- libc-2.27.so
00007fd6433cd000   2048K ----- libc-2.27.so
00007fd6435cd000     16K r---- libc-2.27.so
00007fd6435d1000      8K rw--- libc-2.27.so
```

This is the libc section.

```
00007fd6435d3000     16K rw---   [ anon ]
```

These were the same in step 1 so do not know what they do.

```
00007fd6435d7000    156K r-x-- ld-2.27.so
```

This is the ld section.

```
00007fd6437d3000      8K rw---   [ anon ]
```

These were the same in step 1 so do not know what they do.

```
00007fd6437fe000      4K r---- ld-2.27.so
00007fd6437ff000      4K rw--- ld-2.27.so
```

This is the ld section.

```
00007fd643800000      4K rw---   [ anon ]
```

These were the same in step 1 so do not know what they do.

```
00007ffd6972c000    596K rw---   [ stack ]
```

This is the stack in which variables during runtime are stored.

```
00007ffd697c8000     12K r----   [ anon ]
00007ffd697cb000      4K r-x--   [ anon ]
ffffffffff600000      4K --x--   [ anon ]
```

These were the same in step 1 so do not know what they do.

Step 6:

Output of step 6:

```
0x400000: 7f454c46 0x400004: 02010100 0x400008: 00000000 0x40000c: 00000000
0x400010: 02003e00 0x400014: 01000000 0x400018: 70074000 0x40001c: 00000000
0x400020: 40000000 0x400024: 00000000 0x400028: b02d0000 0x40002c: 00000000
0x400030: 00000000 0x400034: 40003800 0x400038: 09004000 0x40003c: 1e001d00
0x400040: 06000000 0x400044: 04000000 0x400048: 40000000 0x40004c: 00000000
0x400050: 40004000 0x400054: 00000000 0x400058: 40004000 0x40005c: 00000000
0x400060: f8010000 0x400064: 00000000 0x400068: f8010000 0x40006c: 00000000
0x400070: 08000000 0x400074: 00000000 0x400078: 03000000 0x40007c: 04000000
0x400080: 38020000 0x400084: 00000000 0x400088: 38024000 0x40008c: 00000000
0x400090: 38024000 0x400094: 00000000 0x400098: 1c000000 0x40009c: 00000000
0x4000a0: 1c000000 0x4000a4: 00000000 0x4000a8: 01000000 0x4000ac: 00000000
0x4000b0: 01000000 0x4000b4: 05000000 0x4000b8: 00000000 0x4000bc: 00000000
0x4000c0: 00004000 0x4000c4: 00000000 0x4000c8: 00004000 0x4000cc: 00000000
0x4000d0: 10140000 0x4000d4: 00000000 0x4000d8: 10140000 0x4000dc: 00000000
0x4000e0: 00002000 0x4000e4: 00000000 0x4000e8: 01000000 0x4000ec: 06000000
0x4000f0: 081e0000 0x4000f4: 00000000 0x4000f8: 081e6000 0x4000fc: 00000000
0x400100: 081e6000 0x400104: 00000000 0x400108: 18030000 0x40010c: 00000000
0x400110: d8410000 0x400114: 00000000 0x400118: 00002000 0x40011c: 00000000
0x400120: 02000000 0x400124: 06000000 0x400128: 181e0000 0x40012c: 00000000
0x400130: 181e6000 0x400134: 00000000 0x400138: 181e6000 0x40013c: 00000000
0x400140: d0010000 0x400144: 00000000 0x400148: d0010000 0x40014c: 00000000
0x400150: 08000000 0x400154: 00000000 0x400158: 04000000 0x40015c: 04000000
0x400160: 54020000 0x400164: 00000000 0x400168: 54024000 0x40016c: 00000000
0x400170: 54024000 0x400174: 00000000 0x400178: 44000000 0x40017c: 00000000
0x400180: 44000000 0x400184: 00000000 0x400188: 04000000 0x40018c: 00000000
0x400190: 50e57464 0x400194: 04000000 0x400198: 9c110000 0x40019c: 00000000
0x4001a0: 9c114000 0x4001a4: 00000000 0x4001a8: 9c114000 0x4001ac: 00000000
0x4001b0: 7c000000 0x4001b4: 00000000 0x4001b8: 7c000000 0x4001bc: 00000000
0x4001c0: 04000000 0x4001c4: 00000000 0x4001c8: 51e57464 0x4001cc: 06000000
0x4001d0: 00000000 0x4001d4: 00000000 0x4001d8: 00000000 0x4001dc: 00000000
0x4001e0: 00000000 0x4001e4: 00000000 0x4001e8: 00000000 0x4001ec: 00000000
0x4001f0: 00000000 0x4001f4: 00000000 0x4001f8: 10000000 0x4001fc: 00000000
0x400200: 52e57464 0x400204: 04000000 0x400208: 081e0000 0x40020c: 00000000
0x400210: 081e6000 0x400214: 00000000 0x400218: 081e6000 0x40021c: 00000000
0x400220: f8010000 0x400224: 00000000 0x400228: f8010000 0x40022c: 00000000
0x400230: 01000000 0x400234: 00000000 0x400238: 2f6c6962 0x40023c: 36342f6c
0x400240: 642d6c69 0x400244: 6e75782d 0x400248: 7838362d 0x40024c: 36342e73
0x400250: 6f2e3200 0x400254: 04000000 0x400258: 10000000 0x40025c: 01000000
0x400260: 474e5500 0x400264: 00000000 0x400268: 03000000 0x40026c: 02000000
```

```
0x400270: 00000000 0x400274: 04000000 0x400278: 14000000 0x40027c: 03000000
0x400280: 474e5500 0x400284: 6d1e61cc 0x400288: ae03d681 0x40028c: d6c155ee
0x400290: e3f8ac9a 0x400294: def112fa 0x400298: 02000000 0x40029c: 0e000000
0x4002a0: 01000000 0x4002a4: 06000000 0x4002a8: 00000000 0x4002ac: 00400001
0x4002b0: 0e000000 0x4002b4: 00000000 0x4002b8: b92b6b15 0x4002bc: 00000000
0x4002c0: 00000000 0x4002c4: 00000000 0x4002c8: 00000000 0x4002cc: 00000000
0x4002d0: 00000000 0x4002d4: 00000000 0x4002d8: 26000000 0x4002dc: 12000000
0x4002e0: 00000000 0x4002e4: 00000000 0x4002e8: 00000000 0x4002ec: 00000000
0x4002f0: 54000000 0x4002f4: 12000000 0x4002f8: 00000000 0x4002fc: 00000000
0x400300: 00000000 0x400304: 00000000 0x400308: 2b000000 0x40030c: 12000000
0x400310: 00000000 0x400314: 00000000 0x400318: 00000000 0x40031c: 00000000
0x400320: 43000000 0x400324: 12000000 0x400328: 00000000 0x40032c: 00000000
0x400330: 00000000 0x400334: 00000000 0x400338: 0b000000 0x40033c: 12000000
0x400340: 00000000 0x400344: 00000000 0x400348: 00000000 0x40034c: 00000000
0x400350: 4e000000 0x400354: 12000000 0x400358: 00000000 0x40035c: 00000000
0x400360: 00000000 0x400364: 00000000 0x400368: 62000000 0x40036c: 12000000
0x400370: 00000000 0x400374: 00000000 0x400378: 00000000 0x40037c: 00000000
0x400380: 94000000 0x400384: 20000000 0x400388: 00000000 0x40038c: 00000000
0x400390: 00000000 0x400394: 00000000 0x400398: 5b000000 0x40039c: 12000000
0x4003a0: 00000000 0x4003a4: 00000000 0x4003a8: 00000000 0x4003ac: 00000000
0x4003b0: 48000000 0x4003b4: 12000000 0x4003b8: 00000000 0x4003bc: 00000000
0x4003c0: 00000000 0x4003c4: 00000000 0x4003c8: 12000000 0x4003cc: 12000000
0x4003d0: 00000000 0x4003d4: 00000000 0x4003d8: 00000000 0x4003dc: 00000000
0x4003e0: 11000000 0x4003e4: 12000000 0x4003e8: 00000000 0x4003ec: 00000000
0x4003f0: 00000000 0x4003f4: 00000000 0x4003f8: 17000000 0x4003fc: 12000000
0x400400: 00000000 0x400404: 00000000 0x400408: 00000000 0x40040c: 00000000
0x400410: 3c000000 0x400414: 12000000 0x400418: 00000000 0x40041c: 00000000
0x400420: 00000000 0x400424: 00000000 0x400428: 006c6962 0x40042c: 632e736f
0x400430: 2e360068 0x400434: 746f6e6c 0x400438: 00666f70 0x40043c: 656e005f
0x400440: 5f69736f 0x400444: 6339395f 0x400448: 7363616e 0x40044c: 66007075
0x400450: 7473005f 0x400454: 5f737461 0x400458: 636b5f63 0x40045c: 686b5f66
0x400460: 61696c00 0x400464: 7072696e 0x400468: 7466006d 0x40046c: 6d617000
0x400470: 66736565 0x400474: 6b006670 0x400478: 75746300 0x40047c: 66636c6f
0x400480: 7365006d 0x400484: 616c6c6f 0x400488: 63005f5f 0x40048c: 6c696263
0x400490: 5f737461 0x400494: 72745f6d 0x400498: 61696e00 0x40049c: 474c4942
0x4004a0: 435f322e 0x4004a4: 3700474c 0x4004a8: 4942435f 0x4004ac: 322e3400
0x4004b0: 474c4942 0x4004b4: 435f322e 0x4004b8: 322e3500 0x4004bc: 5f5f676d
0x4004c0: 6f6e5f73 0x4004c4: 74617274 0x4004c8: 5f5f0000 0x4004cc: 00000200
0x4004d0: 02000300 0x4004d4: 02000200 0x4004d8: 02000200 0x4004dc: 00000200
```

```
0x4004e0: 02000200 0x4004e4: 02000400 0x4004e8: 02000000 0x4004ec: 00000000
0x4004f0: 01000300 0x4004f4: 01000000 0x4004f8: 10000000 0x4004fc: 00000000
0x400500: 1769690d 0x400504: 00000400 0x400508: 74000000 0x40050c: 10000000
0x400510: 1469690d 0x400514: 00000300 0x400518: 7e000000 0x40051c: 10000000
0x400520: 751a6909 0x400524: 00000200 0x400528: 88000000 0x40052c: 00000000
0x400530: e81f6000 0x400534: 00000000 0x400538: 06000000 0x40053c: 0e000000
0x400540: 00000000 0x400544: 00000000 0x400548: f01f6000 0x40054c: 00000000
0x400550: 06000000 0x400554: 07000000 0x400558: 00000000 0x40055c: 00000000
0x400560: f81f6000 0x400564: 00000000 0x400568: 06000000 0x40056c: 08000000
0x400570: 00000000 0x400574: 00000000 0x400578: 18206000 0x40057c: 00000000
0x400580: 07000000 0x400584: 01000000 0x400588: 00000000 0x40058c: 00000000
0x400590: 20206000 0x400594: 00000000 0x400598: 07000000 0x40059c: 02000000
0x4005a0: 00000000 0x4005a4: 00000000 0x4005a8: 28206000 0x4005ac: 00000000
0x4005b0: 07000000 0x4005b4: 03000000 0x4005b8: 00000000 0x4005bc: 00000000
0x4005c0: 30206000 0x4005c4: 00000000 0x4005c8: 07000000 0x4005cc: 04000000
0x4005d0: 00000000 0x4005d4: 00000000 0x4005d8: 38206000 0x4005dc: 00000000
0x4005e0: 07000000 0x4005e4: 05000000 0x4005e8: 00000000 0x4005ec: 00000000
0x4005f0: 40206000 0x4005f4: 00000000 0x4005f8: 07000000 0x4005fc: 06000000
0x400600: 00000000 0x400604: 00000000 0x400608: 48206000 0x40060c: 00000000
0x400610: 07000000 0x400614: 09000000 0x400618: 00000000 0x40061c: 00000000
0x400620: 50206000 0x400624: 00000000 0x400628: 07000000 0x40062c: 0a000000
0x400630: 00000000 0x400634: 00000000 0x400638: 58206000 0x40063c: 00000000
0x400640: 07000000 0x400644: 0b000000 0x400648: 00000000 0x40064c: 00000000
0x400650: 60206000 0x400654: 00000000 0x400658: 07000000 0x40065c: 0c000000
0x400660: 00000000 0x400664: 00000000 0x400668: 68206000 0x40066c: 00000000
0x400670: 07000000 0x400674: 0d000000 0x400678: 00000000 0x40067c: 00000000
0x400680: 4883ec08 0x400684: 488b056d 0x400688: 19200048 0x40068c: 85c07402
0x400690: ffd04883 0x400694: c408c300 0x400698: 00000000 0x40069c: 00000000
0x4006a0: ff356219 0x4006a4: 2000ff25 0x4006a8: 64192000 0x4006ac: 0f1f4000
0x4006b0: ff256219 0x4006b4: 20006800 0x4006b8: 000000e9 0x4006bc: e0ffffff
0x4006c0: ff255a19 0x4006c4: 20006801 0x4006c8: 000000e9 0x4006cc: d0ffffff
0x4006d0: ff255219 0x4006d4: 20006802 0x4006d8: 000000e9 0x4006dc: c0ffffff
0x4006e0: ff254a19 0x4006e4: 20006803 0x4006e8: 000000e9 0x4006ec: b0ffffff
0x4006f0: ff254219 0x4006f4: 20006804 0x4006f8: 000000e9 0x4006fc: a0ffffff
0x400700: ff253a19 0x400704: 20006805 0x400708: 000000e9 0x40070c: 90ffffff
0x400710: ff253219 0x400714: 20006806 0x400718: 000000e9 0x40071c: 80ffffff
0x400720: ff252a19 0x400724: 20006807 0x400728: 000000e9 0x40072c: 70ffffff
0x400730: ff252219 0x400734: 20006808 0x400738: 000000e9 0x40073c: 60ffffff
0x400740: ff251a19 0x400744: 20006809 0x400748: 000000e9 0x40074c: 50ffffff
```

```
0x400750: ff251219 0x400754: 2000680a 0x400758: 000000e9 0x40075c: 40ffffff
0x400760: ff258218 0x400764: 20006690 0x400768: 00000000 0x40076c: 00000000
0x400770: 31ed4989 0x400774: d15e4889 0x400778: e24883e4 0x40077c: f0505449
0x400780: c7c0000f 0x400784: 400048c7 0x400788: c1900e40 0x40078c: 0048c7c7
0x400790: 57084000 0x400794: ff155618 0x400798: 2000f40f 0x40079c: 1f440000
0x4007a0: f3c3662e 0x4007a4: 0f1f8400 0x4007a8: 00000000 0x4007ac: 0f1f4000
0x4007b0: 55b82021 0x4007b4: 6000483d 0x4007b8: 20216000 0x4007bc: 4889e574
0x4007c0: 17b80000 0x4007c4: 00004885 0x4007c8: c0740d5d 0x4007cc: bf202160
0x4007d0: 00ffe00f 0x4007d4: 1f440000 0x4007d8: 5dc3660f 0x4007dc: 1f440000
0x4007e0: be202160 0x4007e4: 00554881 0x4007e8: ee202160 0x4007ec: 004889e5
0x4007f0: 48c1fe03 0x4007f4: 4889f048 0x4007f8: c1e83f48 0x4007fc: 01c648d1
0x400800: fe7415b8 0x400804: 00000000 0x400808: 4885c074 0x40080c: 0b5dbf20
0x400810: 216000ff 0x400814: e00f1f00 0x400818: 5dc3660f 0x40081c: 1f440000
0x400820: 803df918 0x400824: 20000075 0x400828: 17554889 0x40082c: e5e87eff
0x400830: ffffc605 0x400834: e7182000 0x400838: 015dc30f 0x40083c: 1f440000
0x400840: f3c30f1f 0x400844: 4000662e 0x400848: 0f1f8400 0x40084c: 00000000
0x400850: 554889e5 0x400854: 5deb8955 0x400858: 4889e548 0x40085c: 83ec1064
0x400860: 488b0425 0x400864: 28000000 0x400868: 488945f8 0x40086c: 31c0c645
0x400870: f761b800 0x400874: 000000e8 0x400878: 75010000 0x40087c: eb24488d
0x400880: 3d930600 0x400884: 00e826fe 0x400888: ffff488d 0x40088c: 45f74889
0x400890: c6488d3d 0x400894: 98060000 0x400898: b8000000 0x40089c: 00e8aefe
0x4008a0: ffff0fb6 0x4008a4: 45f73c6e 0x4008a8: 75d4c645 0x4008ac: f761b800
0x4008b0: 000000e8 0x4008b4: ba020000 0x4008b8: eb24488d 0x4008bc: 3d730600
0x4008c0: 00e8eafd 0x4008c4: ffff488d 0x4008c8: 45f74889 0x4008cc: c6488d3d
0x4008d0: 5c060000 0x4008d4: b8000000 0x4008d8: 00e872fe 0x4008dc: ffff0fb6
0x4008e0: 45f73c6e 0x4008e4: 75d4c645 0x4008e8: f761b800 0x4008ec: 000000e8
0x4008f0: dd020000 0x4008f4: eb24488d 0x4008f8: 3d4f0600 0x4008fc: 00e8aefd
0x400900: ffff488d 0x400904: 45f74889 0x400908: c6488d3d 0x40090c: 20060000
0x400910: b8000000 0x400914: 00e836fe 0x400918: ffff0fb6 0x40091c: 45f73c6e
0x400920: 75d4c645 0x400924: f761b800 0x400928: 000000e8 0x40092c: b7020000
0x400930: eb24488d 0x400934: 3d2b0600 0x400938: 00e872fd 0x40093c: ffff488d
0x400940: 45f74889 0x400944: c6488d3d 0x400948: e4050000 0x40094c: b8000000
0x400950: 00e8fafd 0x400954: ffff0fb6 0x400958: 45f73c6e 0x40095c: 75d4c645
0x400960: f761b800 0x400964: 000000e8 0x400968: 30030000 0x40096c: eb24488d
0x400970: 3d070600 0x400974: 00e836fd 0x400978: ffff488d 0x40097c: 45f74889
0x400980: c6488d3d 0x400984: a8050000 0x400988: b8000000 0x40098c: 00e8befd
0x400990: ffff0fb6 0x400994: 45f73c6e 0x400998: 75d4c645 0x40099c: f761b800
0x4009a0: 000000e8 0x4009a4: 23030000 0x4009a8: eb24488d 0x4009ac: 3de30500
0x4009b0: 00e8fafc 0x4009b4: ffff488d 0x4009b8: 45f74889 0x4009bc: c6488d3d
```

```
0x4009c0: 6c050000 0x4009c4: b8000000 0x4009c8: 00e882fd 0x4009cc: ffff0fb6
0x4009d0: 45f73c6e 0x4009d4: 75d4b800 0x4009d8: 00000048 0x4009dc: 8b55f864
0x4009e0: 48331425 0x4009e4: 28000000 0x4009e8: 7405e8e1 0x4009ec: fcffffc9
0x4009f0: c3554889 0x4009f4: e5488d05 0x4009f8: a4162000 0x4009fc: 4889c648
0x400a00: 8d3da505 0x400a04: 0000b800 0x400a08: 000000e8 0x400a0c: 50fdffff
0x400a10: 488d0549 0x400a14: 17200048 0x400a18: 89c6488d 0x400a1c: 3da70500
0x400a20: 00b80000 0x400a24: 0000e835 0x400a28: fdffff48 0x400a2c: 8d050e17
0x400a30: 20004889 0x400a34: c6488d3d 0x400a38: ac050000 0x400a3c: b8000000
0x400a40: 00e81afd 0x400a44: ffff488d 0x400a48: 05fb1620 0x400a4c: 004889c6
0x400a50: 488d3daf 0x400a54: 050000b8 0x400a58: 00000000 0x400a5c: e8fffcff
0x400a60: ff488d05 0x400a64: 0c070000 0x400a68: 4889c648 0x400a6c: 8d3db205
0x400a70: 0000b800 0x400a74: 000000e8 0x400a78: e4fcffff 0x400a7c: 488d05d4
0x400a80: fdffff48 0x400a84: 89c6488d 0x400a88: 3db00500 0x400a8c: 00b80000
0x400a90: 0000e8c9 0x400a94: fcffff48 0x400a98: 8d0553ff 0x400a9c: ffff4889
0x400aa0: c6488d3d 0x400aa4: a7050000 0x400aa8: b8000000 0x400aac: 00e8aefc
0x400ab0: ffff488d 0x400ab4: 05b90000 0x400ab8: 004889c6 0x400abc: 488d3d9f
0x400ac0: 050000b8 0x400ac4: 00000000 0x400ac8: e893fcff 0x400acc: ff488d05
0x400ad0: fd000000 0x400ad4: 4889c648 0x400ad8: 8d3d9705 0x400adc: 0000b800
0x400ae0: 000000e8 0x400ae4: 78fcffff 0x400ae8: 488d05f8 0x400aec: 00000048
0x400af0: 89c6488d 0x400af4: 3d8f0500 0x400af8: 00b80000 0x400afc: 0000e85d
0x400b00: fcffff48 0x400b04: 8d059201 0x400b08: 00004889 0x400b0c: c6488d3d
0x400b10: 87050000 0x400b14: b8000000 0x400b18: 00e842fc 0x400b1c: ffff488d
0x400b20: 05a60100 0x400b24: 004889c6 0x400b28: 488d3d7f 0x400b2c: 050000b8
0x400b30: 00000000 0x400b34: e827fcff 0x400b38: ff48c7c0 0x400b3c: 370e4000
0x400b40: 4889c648 0x400b44: 8d3d7705 0x400b48: 0000b800 0x400b4c: 000000e8
0x400b50: 0cfcffff 0x400b54: 488b058d 0x400b58: 14200048 0x400b5c: 89c6488d
0x400b60: 3d6d0500 0x400b64: 00b80000 0x400b68: 0000e8f1 0x400b6c: fbffff90
0x400b70: 5dc35548 0x400b74: 89e5bf80 0x400b78: 841e00e8 0x400b7c: 90fbffff
0x400b80: 488905b9 0x400b84: 152000bf 0x400b88: 80841e00 0x400b8c: e87ffbff
0x400b90: ff488905 0x400b94: b0152000 0x400b98: 488b05a1 0x400b9c: 15200048
0x400ba0: 89c6488d 0x400ba4: 3d3f0500 0x400ba8: 00b80000 0x400bac: 0000e8ad
0x400bb0: fbffff48 0x400bb4: 8b058e15 0x400bb8: 20004889 0x400bbc: c6488d3d
0x400bc0: 4c050000 0x400bc4: b8000000 0x400bc8: 00e892fb 0x400bcc: ffff905d
0x400bd0: c3554889 0x400bd4: e5bfd430 0x400bd8: 0000b800 0x400bdc: 000000e8
0x400be0: 53020000 0x400be4: 905dc355 0x400be8: 4889e548 0x400bec: 83ec1048
0x400bf0: 8d353c05 0x400bf4: 0000488d 0x400bf8: 3d370500 0x400bfc: 00e83efb
0x400c00: ffff4889 0x400c04: 45f8488b 0x400c08: 45f8ba00 0x400c0c: 000000be
0x400c10: 80841e00 0x400c14: 4889c7e8 0x400c18: 04fbffff 0x400c1c: 488b45f8
0x400c20: 4889c6bf 0x400c24: 00000000 0x400c28: e8d3faff 0x400c2c: ff488b45
```

```
0x400c30: f84889c7 0x400c34: e887faff 0x400c38: ffbe0200 0x400c3c: 0000488d
0x400c40: 3def0400 0x400c44: 00b80000 0x400c48: 0000e8e1 0x400c4c: faffff89
0x400c50: 45f48b45 0x400c54: f441b900 0x400c58: 00000041 0x400c5c: 89c0b902
0x400c60: 000000ba 0x400c64: 03000000 0x400c68: be80841e 0x400c6c: 00bf0000
0x400c70: 0000e869 0x400c74: faffff48 0x400c78: 8905aa14 0x400c7c: 2000488b
0x400c80: 05a31420 0x400c84: 004889c6 0x400c88: 488d3dad 0x400c8c: 040000b8
0x400c90: 00000000 0x400c94: e8c7faff 0x400c98: ff90c9c3 0x400c9c: 554889e5
0x400ca0: c745fc00 0x400ca4: 000000eb 0x400ca8: 16488b15 0x400cac: 78142000
0x400cb0: 8b45fc48 0x400cb4: 984801d0 0x400cb8: c6006c83 0x400cbc: 45fc0181
0x400cc0: 7dfc6f17 0x400cc4: 00007ee1 0x400cc8: 905dc355 0x400ccc: 4889e548
0x400cd0: 83ec1048 0x400cd4: c745f800 0x400cd8: 004000c7 0x400cdc: 45f40000
0x400ce0: 0000e940 0x400ce4: 0100008b 0x400ce8: 45f44898 0x400cec: 488d1485
0x400cf0: 00000000 0x400cf4: 488b45f8 0x400cf8: 4801d08b 0x400cfc: 0089c7e8
0x400d00: ecf9ffff 0x400d04: 89c18b45 0x400d08: f4489848 0x400d0c: 8d148500
0x400d10: 00000048 0x400d14: 8b45f848 0x400d18: 01d089ca 0x400d1c: 4889c648
0x400d20: 8d3d3304 0x400d24: 0000b800 0x400d28: 000000e8 0x400d2c: 30faffff
0x400d30: 8b45f448 0x400d34: 984883c0 0x400d38: 01488d14 0x400d3c: 85000000
0x400d40: 00488b45 0x400d44: f84801d0 0x400d48: 8b0089c7 0x400d4c: e89ff9ff
0x400d50: ff89c18b 0x400d54: 45f44898 0x400d58: 4883c001 0x400d5c: 488d1485
0x400d60: 00000000 0x400d64: 488b45f8 0x400d68: 4801d089 0x400d6c: ca4889c6
0x400d70: 488d3de2 0x400d74: 030000b8 0x400d78: 00000000 0x400d7c: e8dff9ff
0x400d80: ff8b45f4 0x400d84: 48984883 0x400d88: c002488d 0x400d8c: 14850000
0x400d90: 0000488b 0x400d94: 45f84801 0x400d98: d08b0089 0x400d9c: c7e84ef9
0x400da0: ffff89c1 0x400da4: 8b45f448 0x400da8: 984883c0 0x400dac: 02488d14
0x400db0: 85000000 0x400db4: 00488b45 0x400db8: f84801d0 0x400dbc: 89ca4889
0x400dc0: c6488d3d 0x400dc4: 91030000 0x400dc8: b8000000 0x400dcc: 00e88ef9
0x400dd0: ffff8b45 0x400dd4: f4489848 0x400dd8: 83c00348 0x400ddc: 8d148500
0x400de0: 00000048 0x400de4: 8b45f848 0x400de8: 01d08b00 0x400dec: 89c7e8fd
0x400df0: f8ffff89 0x400df4: c18b45f4 0x400df8: 48984883 0x400dfc: c003488d
0x400e00: 14850000 0x400e04: 0000488b 0x400e08: 45f84801 0x400e0c: d089ca48
0x400e10: 89c6488d 0x400e14: 3d4d0300 0x400e18: 00b80000 0x400e1c: 0000e83d
0x400e20: f9ffff83 0x400e24: 45f40481 0x400e28: 7df4fb03 0x400e2c: 00000f8e
0x400e30: b3feffff 0x400e34: 90c9c355 0x400e38: 4889e548 0x400e3c: 83ec2089
0x400e40: 7dec8b45 0x400e44: ec85c07e 0x400e48: 4048c745 0x400e4c: f8010000
0x400e50: 008b45ec 0x400e54: 3dd43000 0x400e58: 0074088b 0x400e5c: 45ec83f8
0x400e60: 01751848 0x400e64: 8d45ec48 0x400e68: 89c6488d 0x400e6c: 3d070300
0x400e70: 00b80000 0x400e74: 0000e8e5 0x400e78: f8ffff8b 0x400e7c: 45ec83e8
0x400e80: 0189c7e8 0x400e84: afffffff 0x400e88: 9090c9c3 0x400e8c: 0f1f4000
0x400e90: 41574156 0x400e94: 4989d741 0x400e98: 5541544c 0x400e9c: 8d25660f
```

```
0x400ea0: 20005548 0x400ea4: 8d2d660f 0x400ea8: 20005341 0x400eac: 89fd4989
0x400eb0: f64c29e5 0x400eb4: 4883ec08 0x400eb8: 48c1fd03 0x400ebc: e8bff7ff
0x400ec0: ff4885ed 0x400ec4: 742031db 0x400ec8: 0f1f8400 0x400ecc: 00000000
0x400ed0: 4c89fa4c 0x400ed4: 89f64489 0x400ed8: ef41ff14 0x400edc: dc4883c3
0x400ee0: 014839dd 0x400ee4: 75ea4883 0x400ee8: c4085b5d 0x400eec: 415c415d
0x400ef0: 415e415f 0x400ef4: c390662e 0x400ef8: 0f1f8400 0x400efc: 00000000
0x400f00: f3c30000 0x400f04: 4883ec08 0x400f08: 4883c408 0x400f0c: c3000000
0x400f10: 01000200 0x400f14: 00000000 0x400f18: 50726573 0x400f1c: 73206e20
0x400f20: 746f2067 0x400f24: 6f20746f 0x400f28: 20537465 0x400f2c: 70203200
0x400f30: 20256300 0x400f34: 50726573 0x400f38: 73206e20 0x400f3c: 746f2067
0x400f40: 6f20746f 0x400f44: 20537465 0x400f48: 70203300 0x400f4c: 50726573
0x400f50: 73206e20 0x400f54: 746f2067 0x400f58: 6f20746f 0x400f5c: 20537465
0x400f60: 70203400 0x400f64: 50726573 0x400f68: 73206e20 0x400f6c: 746f2067
0x400f70: 6f20746f 0x400f74: 20537465 0x400f78: 70203500 0x400f7c: 50726573
0x400f80: 73206e20 0x400f84: 746f2067 0x400f88: 6f20746f 0x400f8c: 20537465
0x400f90: 70203600 0x400f94: 50726573 0x400f98: 73206e20 0x400f9c: 746f2065
0x400fa0: 6e642070 0x400fa4: 726f6772 0x400fa8: 616d0069 0x400fac: 6e697469
0x400fb0: 616c697a 0x400fb4: 65642061 0x400fb8: 64647265 0x400fbc: 73733a20
0x400fc0: 3078256c 0x400fc4: 78200a00 0x400fc8: 756e5f69 0x400fcc: 6e697469
0x400fd0: 616c697a 0x400fd4: 65642061 0x400fd8: 64647265 0x400fdc: 73733a20
0x400fe0: 3078256c 0x400fe4: 78200a00 0x400fe8: 756e5f69 0x400fec: 6e69745f
```

Since the output of the main in objdump of app is :

```
0000000000400857 <main>:
  40087c:    eb 24                  jmp    4008a2 <main+0x4b>
  4008a8:    75 d4                  jne    40087e <main+0x27>
  4008b8:    eb 24                  jmp    4008de <main+0x87>
  4008e4:    75 d4                  jne    4008ba <main+0x63>
  4008f4:    eb 24                  jmp    40091a <main+0xc3>
  400920:    75 d4                  jne    4008f6 <main+0x9f>
  400930:    eb 24                  jmp    400956 <main+0xff>
  40095c:    75 d4                  jne    400932 <main+0xdb>
  40096c:    eb 24                  jmp    400992 <main+0x13b>
  400998:    75 d4                  jne    40096e <main+0x117>
  4009a8:    eb 24                  jmp    4009ce <main+0x177>
  4009d4:    75 d4                  jne    4009aa <main+0x153>
  4009e8:    74 05                  je     4009ef <main+0x198>
  400a7c:    48 8d 05 d4 fd ff ff   lea    -0x22c(%rip),%rax        # 400857 <main>
```

At 40087c objdump is showing the first instruction in the main function which is

```
 40087c:          eb 24                        jmp    4008a2 <main+0x4b>
```

Then trying to find 40087c in our output, I can see that,

```
0x40087c: eb 24 48 8d
```

Thus we can clearly see that the output of our program shows our main function and all the instruction in it.

By compiling the app with -static flag, we can see that the libc and all libraries are included in the executable. Thus increasing the size of the executable from 13.6 KB to 963.2 KB. Thus including all the libc code in the objdump as well. Since the new objdump is 3000 pages long, I will not be including that in my document but since libc and libraries were added statically to the executable, the extra size of the objdump has the code of these libraries.

PMAP at step 1:

```
23948:   ./app
0000000000400000      8K r-x-- app
0000000000601000      4K r---- app
0000000000602000      4K rw--- app
0000000000603000     12K rw---   [ anon ]
0000000002508000    132K rw---   [ anon ]
00007fd6431e6000   1948K r-x-- libc-2.27.so
00007fd6433cd000   2048K ----- libc-2.27.so
00007fd6435cd000     16K r---- libc-2.27.so
00007fd6435d1000      8K rw--- libc-2.27.so
00007fd6435d3000     16K rw---   [ anon ]
00007fd6435d7000    156K r-x-- ld-2.27.so
00007fd6437d3000      8K rw---   [ anon ]
00007fd6437fe000      4K r---- ld-2.27.so
00007fd6437ff000      4K rw--- ld-2.27.so
00007fd643800000      4K rw---   [ anon ]
00007ffd697a0000    132K rw---   [ stack ]
00007ffd697c8000     12K r----   [ anon ]
00007ffd697cb000      4K r-x--   [ anon ]
ffffffffff600000      4K --x--   [ anon ]
 total             4524K
```

PMAP at step 2:

```
23948:   ./app
Address           Kbytes     RSS   Dirty Mode  Mapping
0000000000400000       8       8       0 r-x-- app
0000000000400000       0       0       0 r-x-- app
0000000000601000       4       4       4 r---- app
0000000000601000       0       0       0 r---- app
0000000000602000       4       4       4 rw--- app
0000000000602000       0       0       0 rw--- app
0000000000603000      12       0       0 rw---   [ anon ]
0000000000603000       0       0       0 rw---   [ anon ]
0000000002508000     132       4       4 rw---   [ anon ]
0000000002508000       0       0       0 rw---   [ anon ]
00007fd642e14000    3912       8       8 rw---   [ anon ]
00007fd642e14000       0       0       0 rw---   [ anon ]
00007fd6431e6000    1948    1240       0 r-x-- libc-2.27.so
```

```
00007fd6431e6000       0       0       0 r-x-- libc-2.27.so
00007fd6433cd000    2048       0       0 ----- libc-2.27.so
00007fd6433cd000       0       0       0 ----- libc-2.27.so
00007fd6435cd000      16      16      16 r---- libc-2.27.so
00007fd6435cd000       0       0       0 r---- libc-2.27.so
00007fd6435d1000       8       8       8 rw--- libc-2.27.so
00007fd6435d1000       0       0       0 rw--- libc-2.27.so
00007fd6435d3000      16      12      12 rw---   [ anon ]
00007fd6435d3000       0       0       0 rw---   [ anon ]
00007fd6435d7000     156     156       0 r-x-- ld-2.27.so
00007fd6435d7000       0       0       0 r-x-- ld-2.27.so
00007fd6437d3000       8       8       8 rw---   [ anon ]
00007fd6437d3000       0       0       0 rw---   [ anon ]
00007fd6437fe000       4       4       4 r---- ld-2.27.so
00007fd6437fe000       0       0       0 r---- ld-2.27.so
00007fd6437ff000       4       4       4 rw--- ld-2.27.so
00007fd6437ff000       0       0       0 rw--- ld-2.27.so
00007fd643800000       4       4       4 rw---   [ anon ]
00007fd643800000       0       0       0 rw---   [ anon ]
00007ffd697a0000     132      12      12 rw---   [ stack ]
00007ffd697a0000       0       0       0 rw---   [ stack ]
00007ffd697c8000      12       0       0 r----   [ anon ]
00007ffd697c8000       0       0       0 r----   [ anon ]
00007ffd697cb000       4       4       0 r-x--   [ anon ]
00007ffd697cb000       0       0       0 r-x--   [ anon ]
ffffffffff600000       4       0       0 --x--   [ anon ]
ffffffffff600000       0       0       0 --x--   [ anon ]
---------------- ------- ------- -------
total kB            8436    1496      88
```

PMAP at step 3:

```
23948:   ./app
Address           Kbytes     RSS   Dirty Mode  Mapping
0000000000400000       8       8       0 r-x-- app
0000000000400000       0       0       0 r-x-- app
0000000000601000       4       4       4 r---- app
0000000000601000       0       0       0 r---- app
0000000000602000       4       4       4 rw--- app
0000000000602000       0       0       0 rw--- app
```

```
0000000000603000      12       0       0 rw---   [ anon ]
0000000000603000       0       0       0 rw---   [ anon ]
0000000002508000     132       4       4 rw---   [ anon ]
0000000002508000       0       0       0 rw---   [ anon ]
00007fd642e14000    3912       8       8 rw---   [ anon ]
00007fd642e14000       0       0       0 rw---   [ anon ]
00007fd6431e6000    1948    1240       0 r-x-- libc-2.27.so
00007fd6431e6000       0       0       0 r-x-- libc-2.27.so
00007fd6433cd000    2048       0       0 ----- libc-2.27.so
00007fd6433cd000       0       0       0 ----- libc-2.27.so
00007fd6435cd000      16      16      16 r---- libc-2.27.so
00007fd6435cd000       0       0       0 r---- libc-2.27.so
00007fd6435d1000       8       8       8 rw--- libc-2.27.so
00007fd6435d1000       0       0       0 rw--- libc-2.27.so
00007fd6435d3000      16      12      12 rw---   [ anon ]
00007fd6435d3000       0       0       0 rw---   [ anon ]
00007fd6435d7000     156     156       0 r-x-- ld-2.27.so
00007fd6435d7000       0       0       0 r-x-- ld-2.27.so
00007fd6437d3000       8       8       8 rw---   [ anon ]
00007fd6437d3000       0       0       0 rw---   [ anon ]
00007fd6437fe000       4       4       4 r---- ld-2.27.so
00007fd6437fe000       0       0       0 r---- ld-2.27.so
00007fd6437ff000       4       4       4 rw--- ld-2.27.so
00007fd6437ff000       0       0       0 rw--- ld-2.27.so
00007fd643800000       4       4       4 rw---   [ anon ]
00007fd643800000       0       0       0 rw---   [ anon ]
00007ffd6972c000     596     596     596 rw---   [ stack ]
00007ffd6972c000       0       0       0 rw---   [ stack ]
00007ffd697c8000      12       0       0 r----   [ anon ]
00007ffd697c8000       0       0       0 r----   [ anon ]
00007ffd697cb000       4       4       0 r-x--   [ anon ]
00007ffd697cb000       0       0       0 r-x--   [ anon ]
ffffffffff600000       4       0       0 --x--   [ anon ]
ffffffffff600000       0       0       0 --x--   [ anon ]
---------------- ------- ------- -------
total kB            8900    2080     672
```

PMAP at step 4:

```
23948:   ./app
```

```
Address           Kbytes     RSS   Dirty Mode  Mapping
0000000000400000       8       8       0 r-x-- app
0000000000400000       0       0       0 r-x-- app
0000000000601000       4       4       4 r---- app
0000000000601000       0       0       0 r---- app
0000000000602000       4       4       4 rw--- app
0000000000602000       0       0       0 rw--- app
0000000000603000      12       0       0 rw---   [ anon ]
0000000000603000       0       0       0 rw---   [ anon ]
0000000002508000     132       8       8 rw---   [ anon ]
0000000002508000       0       0       0 rw---   [ anon ]
00007fd642c2b000    1956       0       0 rw--- shm.txt
00007fd642c2b000       0       0       0 rw--- shm.txt
00007fd642e14000    3912       8       8 rw---   [ anon ]
00007fd642e14000       0       0       0 rw---   [ anon ]
00007fd6431e6000    1948    1240       0 r-x-- libc-2.27.so
00007fd6431e6000       0       0       0 r-x-- libc-2.27.so
00007fd6433cd000    2048       0       0 ----- libc-2.27.so
00007fd6433cd000       0       0       0 ----- libc-2.27.so
00007fd6435cd000      16      16      16 r---- libc-2.27.so
00007fd6435cd000       0       0       0 r---- libc-2.27.so
00007fd6435d1000       8       8       8 rw--- libc-2.27.so
00007fd6435d1000       0       0       0 rw--- libc-2.27.so
00007fd6435d3000      16      12      12 rw---   [ anon ]
00007fd6435d3000       0       0       0 rw---   [ anon ]
00007fd6435d7000     156     156       0 r-x-- ld-2.27.so
00007fd6435d7000       0       0       0 r-x-- ld-2.27.so
00007fd6437d3000       8       8       8 rw---   [ anon ]
00007fd6437d3000       0       0       0 rw---   [ anon ]
00007fd6437fe000       4       4       4 r---- ld-2.27.so
00007fd6437fe000       0       0       0 r---- ld-2.27.so
00007fd6437ff000       4       4       4 rw--- ld-2.27.so
00007fd6437ff000       0       0       0 rw--- ld-2.27.so
00007fd643800000       4       4       4 rw---   [ anon ]
00007fd643800000       0       0       0 rw---   [ anon ]
00007ffd6972c000     596     596     596 rw---   [ stack ]
00007ffd6972c000       0       0       0 rw---   [ stack ]
00007ffd697c8000      12       0       0 r----   [ anon ]
00007ffd697c8000       0       0       0 r----   [ anon ]
```

```
00007ffd697cb000       4       4       0 r-x--   [ anon ]
00007ffd697cb000       0       0       0 r-x--   [ anon ]
ffffffffff600000       4       0       0 --x--   [ anon ]
ffffffffff600000       0       0       0 --x--   [ anon ]
---------------- ------- ------- -------
total kB           10856    2084     676
```

PMAP at step 5:

```
23948:   ./app
Address           Kbytes     RSS   Dirty Mode  Mapping
0000000000400000       8       8       0 r-x-- app
0000000000400000       0       0       0 r-x-- app
0000000000601000       4       4       4 r---- app
0000000000601000       0       0       0 r---- app
0000000000602000       4       4       4 rw--- app
0000000000602000       0       0       0 rw--- app
0000000000603000      12       0       0 rw---   [ anon ]
0000000000603000       0       0       0 rw---   [ anon ]
0000000002508000     132       8       8 rw---   [ anon ]
0000000002508000       0       0       0 rw---   [ anon ]
00007fd642c2b000    1956       8       8 rw--- shm.txt
00007fd642c2b000       0       0       0 rw--- shm.txt
00007fd642e14000    3912       8       8 rw---   [ anon ]
00007fd642e14000       0       0       0 rw---   [ anon ]
00007fd6431e6000    1948    1240       0 r-x-- libc-2.27.so
00007fd6431e6000       0       0       0 r-x-- libc-2.27.so
00007fd6433cd000    2048       0       0 ----- libc-2.27.so
00007fd6433cd000       0       0       0 ----- libc-2.27.so
00007fd6435cd000      16      16      16 r---- libc-2.27.so
00007fd6435cd000       0       0       0 r---- libc-2.27.so
00007fd6435d1000       8       8       8 rw--- libc-2.27.so
00007fd6435d1000       0       0       0 rw--- libc-2.27.so
00007fd6435d3000      16      12      12 rw---   [ anon ]
00007fd6435d3000       0       0       0 rw---   [ anon ]
00007fd6435d7000     156     156       0 r-x-- ld-2.27.so
00007fd6435d7000       0       0       0 r-x-- ld-2.27.so
00007fd6437d3000       8       8       8 rw---   [ anon ]
00007fd6437d3000       0       0       0 rw---   [ anon ]
00007fd6437fe000       4       4       4 r---- ld-2.27.so
```

```
00007fd6437fe000       0       0       0 r---- ld-2.27.so
00007fd6437ff000       4       4       4 rw--- ld-2.27.so
00007fd6437ff000       0       0       0 rw--- ld-2.27.so
00007fd643800000       4       4       4 rw---   [ anon ]
00007fd643800000       0       0       0 rw---   [ anon ]
00007ffd6972c000     596     596     596 rw---   [ stack ]
00007ffd6972c000       0       0       0 rw---   [ stack ]
00007ffd697c8000      12       0       0 r----   [ anon ]
00007ffd697c8000       0       0       0 r----   [ anon ]
00007ffd697cb000       4       4       0 r-x--   [ anon ]
00007ffd697cb000       0       0       0 r-x--   [ anon ]
ffffffffff600000       4       0       0 --x--   [ anon ]
ffffffffff600000       0       0       0 --x--   [ anon ]
---------------- ------- ------- -------
total kB           10856    2092     684
```

PMAP at step 6:

```
23948:   ./app
0000000000400000      8K r-x-- app
0000000000601000      4K r---- app
0000000000602000      4K rw--- app
0000000000603000     12K rw---   [ anon ]
0000000002508000    132K rw---   [ anon ]
00007fd642c2b000   1956K rw--- shm.txt
00007fd642e14000   3912K rw---   [ anon ]
00007fd6431e6000   1948K r-x-- libc-2.27.so
00007fd6433cd000   2048K ----- libc-2.27.so
00007fd6435cd000     16K r---- libc-2.27.so
00007fd6435d1000      8K rw--- libc-2.27.so
00007fd6435d3000     16K rw---   [ anon ]
00007fd6435d7000    156K r-x-- ld-2.27.so
00007fd6437d3000      8K rw---   [ anon ]
00007fd6437fe000      4K r---- ld-2.27.so
00007fd6437ff000      4K rw--- ld-2.27.so
00007fd643800000      4K rw---   [ anon ]
00007ffd6972c000    596K rw---   [ stack ]
00007ffd697c8000     12K r----   [ anon ]
00007ffd697cb000      4K r-x--   [ anon ]
ffffffffff600000      4K --x--   [ anon ]
```

```
 total           10856K
```

OBJDUMP of app:

```
app:     file format elf64-x86-64
app
architecture: i386:x86-64, flags 0x00000112:
EXEC_P, HAS_SYMS, D_PAGED
start address 0x0000000000400770

Program Header:
    PHDR off    0x0000000000000040 vaddr 0x0000000000400040 paddr 0x0000000000400040
align 2**3
         filesz 0x00000000000001f8 memsz 0x00000000000001f8 flags r--
  INTERP off    0x0000000000000238 vaddr 0x0000000000400238 paddr 0x0000000000400238
align 2**0
         filesz 0x000000000000001c memsz 0x000000000000001c flags r--
    LOAD off    0x0000000000000000 vaddr 0x0000000000400000 paddr 0x0000000000400000
align 2**21
         filesz 0x0000000000001318 memsz 0x0000000000001318 flags r-x
    LOAD off    0x0000000000001e08 vaddr 0x0000000000601e08 paddr 0x0000000000601e08
align 2**21
         filesz 0x0000000000000318 memsz 0x00000000000041d8 flags rw-
 DYNAMIC off    0x0000000000001e18 vaddr 0x0000000000601e18 paddr
0x0000000000601e18 align 2**3
         filesz 0x00000000000001d0 memsz 0x00000000000001d0 flags rw-
    NOTE off    0x0000000000000254 vaddr 0x0000000000400254 paddr 0x0000000000400254
align 2**2
         filesz 0x0000000000000044 memsz 0x0000000000000044 flags r--
EH_FRAME off    0x00000000000010a4 vaddr 0x00000000004010a4 paddr
0x00000000004010a4 align 2**2
         filesz 0x000000000000007c memsz 0x000000000000007c flags r--
   STACK off    0x0000000000000000 vaddr 0x0000000000000000 paddr 0x0000000000000000
align 2**4
         filesz 0x0000000000000000 memsz 0x0000000000000000 flags rw-
   RELRO off    0x0000000000001e08 vaddr 0x0000000000601e08 paddr 0x0000000000601e08
align 2**0
         filesz 0x00000000000001f8 memsz 0x00000000000001f8 flags r--

Dynamic Section:
  NEEDED               libc.so.6
  INIT                 0x0000000000400680
  FINI                 0x0000000000400e14
  INIT_ARRAY           0x0000000000601e08
  INIT_ARRAYSZ         0x0000000000000008
  FINI_ARRAY           0x0000000000601e10
  FINI_ARRAYSZ         0x0000000000000008
  GNU_HASH             0x0000000000400298
  STRTAB               0x0000000000400428
  SYMTAB               0x00000000004002c0
  STRSZ                0x00000000000000a3
  SYMENT               0x0000000000000018
  DEBUG                0x0000000000000000
  PLTGOT               0x0000000000602000
  PLTRELSZ             0x0000000000000108
  PLTREL               0x0000000000000007
  JMPREL               0x0000000000400578
  RELA                 0x0000000000400530
  RELASZ               0x0000000000000048
  RELAENT              0x0000000000000018
  VERNEED              0x00000000004004f0
  VERNEEDNUM           0x0000000000000001
  VERSYM               0x00000000004004cc

Version References:
  required from libc.so.6:
    0x0d696917 0x00 04 GLIBC_2.7
    0x0d696914 0x00 03 GLIBC_2.4
    0x09691a75 0x00 02 GLIBC_2.2.5

Sections:
Idx Name          Size      VMA               LMA               File off  Algn
  0 .interp       0000001c  0000000000400238  0000000000400238  00000238  2**0
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  1 .note.ABI-tag 00000020  0000000000400254  0000000000400254  00000254  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  2 .note.gnu.build-id 00000024  0000000000400274  0000000000400274  00000274  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  3 .gnu.hash     00000024  0000000000400298  0000000000400298  00000298  2**3
```

```
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .dynsym       00000168  00000000004002c0  00000000004002c0  000002c0  2**3
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  5 .dynstr       000000a3  0000000000400428  0000000000400428  00000428  2**0
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  6 .gnu.version  0000001e  00000000004004cc  00000000004004cc  000004cc  2**1
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  7 .gnu.version_r 00000040  00000000004004f0  00000000004004f0  000004f0  2**3
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  8 .rela.dyn     00000048  0000000000400530  0000000000400530  00000530  2**3
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  9 .rela.plt     00000108  0000000000400578  0000000000400578  00000578  2**3
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
 10 .init         00000017  0000000000400680  0000000000400680  00000680  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
 11 .plt          000000c0  00000000004006a0  00000000004006a0  000006a0  2**4
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
 12 .plt.got      00000008  0000000000400760  0000000000400760  00000760  2**3
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
 13 .text         000006a2  0000000000400770  0000000000400770  00000770  2**4
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
 14 .fini         00000009  0000000000400e14  0000000000400e14  00000e14  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
 15 .rodata       00000283  0000000000400e20  0000000000400e20  00000e20  2**3
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
 16 .eh_frame_hdr 0000007c  00000000004010a4  00000000004010a4  000010a4  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
 17 .eh_frame     000001f8  0000000000401120  0000000000401120  00001120  2**3
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
 18 .init_array   00000008  0000000000601e08  0000000000601e08  00001e08  2**3
                  CONTENTS, ALLOC, LOAD, DATA
 19 .fini_array   00000008  0000000000601e10  0000000000601e10  00001e10  2**3
                  CONTENTS, ALLOC, LOAD, DATA
 20 .dynamic      000001d0  0000000000601e18  0000000000601e18  00001e18  2**3
                  CONTENTS, ALLOC, LOAD, DATA
 21 .got          00000018  0000000000601fe8  0000000000601fe8  00001fe8  2**3
                  CONTENTS, ALLOC, LOAD, DATA
 22 .got.plt      00000070  0000000000602000  0000000000602000  00002000  2**3
                  CONTENTS, ALLOC, LOAD, DATA
 23 .data         000000a0  0000000000602080  0000000000602080  00002080  2**5
                  CONTENTS, ALLOC, LOAD, DATA
 24 .bss          00003ec0  0000000000602120  0000000000602120  00002120  2**5
                  ALLOC
 25 .comment      00000029  0000000000000000  0000000000000000  00002120  2**0
                  CONTENTS, READONLY
SYMBOL TABLE:
0000000000400238 l    d  .interp	0000000000000000              .interp
0000000000400254 l    d  .note.ABI-tag	0000000000000000              .note.ABI-tag
0000000000400274 l    d  .note.gnu.build-id	0000000000000000              .note.gnu.build-id
0000000000400298 l    d  .gnu.hash	0000000000000000              .gnu.hash
00000000004002c0 l    d  .dynsym	0000000000000000              .dynsym
0000000000400428 l    d  .dynstr	0000000000000000              .dynstr
00000000004004cc l    d  .gnu.version	0000000000000000              .gnu.version
00000000004004f0 l    d  .gnu.version_r	0000000000000000              .gnu.version_r
0000000000400530 l    d  .rela.dyn	0000000000000000              .rela.dyn
0000000000400578 l    d  .rela.plt	0000000000000000              .rela.plt
0000000000400680 l    d  .init	0000000000000000              .init
00000000004006a0 l    d  .plt	0000000000000000              .plt
0000000000400760 l    d  .plt.got	0000000000000000              .plt.got
0000000000400770 l    d  .text	0000000000000000              .text
0000000000400e14 l    d  .fini	0000000000000000              .fini
0000000000400e20 l    d  .rodata	0000000000000000              .rodata
00000000004010a4 l    d  .eh_frame_hdr	0000000000000000              .eh_frame_hdr
0000000000401120 l    d  .eh_frame	0000000000000000              .eh_frame
0000000000601e08 l    d  .init_array	0000000000000000              .init_array
0000000000601e10 l    d  .fini_array	0000000000000000              .fini_array
0000000000601e18 l    d  .dynamic	0000000000000000              .dynamic
0000000000601fe8 l    d  .got	0000000000000000              .got
0000000000602000 l    d  .got.plt	0000000000000000              .got.plt
0000000000602080 l    d  .data	0000000000000000              .data
0000000000602120 l    d  .bss	0000000000000000              .bss
0000000000000000 l    d  .comment	0000000000000000              .comment
0000000000000000 l    df *ABS*	0000000000000000              crtstuff.c
00000000004007b0 l     F .text	0000000000000000              deregister_tm_clones
00000000004007e0 l     F .text	0000000000000000              register_tm_clones
0000000000400820 l     F .text	0000000000000000              __do_global_dtors_aux
0000000000602120 l     O .bss	0000000000000001              completed.7698
```

```
0000000000601e10 l     O .fini_array 0000000000000000
__do_global_dtors_aux_fini_array_entry
0000000000400850 l     F .text          0000000000000000            frame_dummy
0000000000601e08 l     O .init_array 0000000000000000
__frame_dummy_init_array_entry
0000000000000000 l    df *ABS*      0000000000000000            module1.c
0000000000000000 l    df *ABS*      0000000000000000            module2.c
0000000000000000 l    df *ABS*      0000000000000000            module3.c
0000000000000000 l    df *ABS*      0000000000000000            crtstuff.c
0000000000401314 l     O .eh_frame 0000000000000000            __FRAME_END__
0000000000000000 l    df *ABS*      0000000000000000
0000000000601e10 l       .init_array  0000000000000000            __init_array_end
0000000000601e18 l     O .dynamic  0000000000000000            _DYNAMIC
0000000000601e08 l       .init_array  0000000000000000            __init_array_start
00000000004010a4 l       .eh_frame_hdr       0000000000000000
__GNU_EH_FRAME_HDR
0000000000602000 l     O .got.plt     0000000000000000
_GLOBAL_OFFSET_TABLE_
0000000000400e10 g     F .text         0000000000000002            __libc_csu_fini
0000000000602140 g     O .bss          0000000000000008            un_init_ptr1
0000000000400c9c g     F .text         000000000000002f            step5
0000000000400bd1 g     F .text         0000000000000016            step3
00000000004009f1 g     F .text         0000000000000181            step1
0000000000602080  w      .data         0000000000000000            data_start
0000000000000000       F *UND*     0000000000000000            puts@@GLIBC_2.2.5
0000000000602120 g       .data         0000000000000000            _edata
0000000000000000       F *UND*     0000000000000000            fclose@@GLIBC_2.2.5
0000000000602128 g     O .bss          0000000000000008            region
0000000000400e14 g     F .fini          0000000000000000            _fini
0000000000000000       F *UND*     0000000000000000
__stack_chk_fail@@GLIBC_2.4
0000000000000000       F *UND*     0000000000000000            mmap@@GLIBC_2.2.5
0000000000000000       F *UND*     0000000000000000            printf@@GLIBC_2.2.5
0000000000000000       F *UND*     0000000000000000            htonl@@GLIBC_2.2.5
0000000000000000       F *UND*     0000000000000000            fputc@@GLIBC_2.2.5
0000000000000000       F *UND*     0000000000000000
__libc_start_main@@GLIBC_2.2.5
0000000000602080 g       .data         0000000000000000            __data_start
0000000000000000  w      *UND*     0000000000000000            __gmon_start__
```

```
0000000000602088 g     O .data     0000000000000000          .hidden __dso_handle
00000000006020a0 g     O .data     0000000000000080          initialized
0000000000400e20 g     O .rodata   0000000000000004          _IO_stdin_used
0000000000602148 g     O .bss      0000000000000008          un_init_ptr2
0000000000602160 g     O .bss      0000000000003e80          un_initialized
0000000000400be7 g     F .text     00000000000000b5          step4
0000000000400b72 g     F .text     000000000000005f          step2
0000000000400da0 g     F .text     0000000000000065          __libc_csu_init
0000000000400ccb g     F .text     0000000000000072          step6
0000000000000000       F *UND*     0000000000000000          malloc@@GLIBC_2.2.5
0000000000400d3d g     F .text     0000000000000055          foo
0000000000605fe0 g       .bss 0000000000000000          _end
00000000004007a0 g     F .text     0000000000000002          .hidden _dl_relocate_static_pie
0000000000400770 g     F .text     000000000000002b          _start
0000000000000000       F *UND*     0000000000000000          fseek@@GLIBC_2.2.5
0000000000602120 g       .bss      0000000000000000          __bss_start
0000000000400857 g     F .text     000000000000019a          main
0000000000401078 g     O .rodata   0000000000000004          init_const
0000000000000000       F *UND*     0000000000000000          open@@GLIBC_2.2.5
0000000000000000       F *UND*     0000000000000000          fopen@@GLIBC_2.2.5
0000000000000000       F *UND*     0000000000000000
__isoc99_scanf@@GLIBC_2.7
0000000000602120 g     O .data     0000000000000000          .hidden __TMC_END__
0000000000400680 g     F .init     0000000000000000          _init



Disassembly of section .init:

0000000000400680 <_init>:
  400680:       48 83 ec 08             sub    $0x8,%rsp
  400684:       48 8b 05 6d 19 20 00    mov    0x20196d(%rip),%rax        # 601ff8
<__gmon_start__>
  40068b:       48 85 c0                test   %rax,%rax
  40068e:       74 02                   je     400692 <_init+0x12>
  400690:       ff d0             callq  *%rax
  400692:       48 83 c4 08             add    $0x8,%rsp
  400696:       c3                retq
```

```
Disassembly of section .plt:

00000000004006a0 <.plt>:
  4006a0:       ff 35 62 19 20 00       pushq  0x201962(%rip)        # 602008
<_GLOBAL_OFFSET_TABLE_+0x8>
  4006a6:       ff 25 64 19 20 00       jmpq   *0x201964(%rip)        # 602010
<_GLOBAL_OFFSET_TABLE_+0x10>
  4006ac:       0f 1f 40 00             nopl   0x0(%rax)

00000000004006b0 <puts@plt>:
  4006b0:       ff 25 62 19 20 00       jmpq   *0x201962(%rip)        # 602018
<puts@GLIBC_2.2.5>
  4006b6:       68 00 00 00 00          pushq  $0x0
  4006bb:       e9 e0 ff ff ff          jmpq   4006a0 <.plt>

00000000004006c0 <fclose@plt>:
  4006c0:       ff 25 5a 19 20 00       jmpq   *0x20195a(%rip)        # 602020
<fclose@GLIBC_2.2.5>
  4006c6:       68 01 00 00 00          pushq  $0x1
  4006cb:       e9 d0 ff ff ff          jmpq   4006a0 <.plt>

00000000004006d0 <__stack_chk_fail@plt>:
  4006d0:       ff 25 52 19 20 00       jmpq   *0x201952(%rip)        # 602028
<__stack_chk_fail@GLIBC_2.4>
  4006d6:       68 02 00 00 00          pushq  $0x2
  4006db:       e9 c0 ff ff ff          jmpq   4006a0 <.plt>

00000000004006e0 <mmap@plt>:
  4006e0:       ff 25 4a 19 20 00       jmpq   *0x20194a(%rip)        # 602030
<mmap@GLIBC_2.2.5>
  4006e6:       68 03 00 00 00          pushq  $0x3
  4006eb:       e9 b0 ff ff ff          jmpq   4006a0 <.plt>

00000000004006f0 <htonl@plt>:
  4006f0:       ff 25 42 19 20 00       jmpq   *0x201942(%rip)        # 602038
<htonl@GLIBC_2.2.5>
  4006f6:       68 04 00 00 00          pushq  $0x4
  4006fb:       e9 a0 ff ff ff          jmpq   4006a0 <.plt>
```

```
0000000000400700 <fputc@plt>:
  400700:       ff 25 3a 19 20 00       jmpq   *0x20193a(%rip)        # 602040
<fputc@GLIBC_2.2.5>
  400706:       68 05 00 00 00          pushq  $0x5
  40070b:       e9 90 ff ff ff          jmpq   4006a0 <.plt>

0000000000400710 <malloc@plt>:
  400710:       ff 25 32 19 20 00       jmpq   *0x201932(%rip)        # 602048
<malloc@GLIBC_2.2.5>
  400716:       68 06 00 00 00          pushq  $0x6
  40071b:       e9 80 ff ff ff          jmpq   4006a0 <.plt>

0000000000400720 <fseek@plt>:
  400720:       ff 25 2a 19 20 00       jmpq   *0x20192a(%rip)        # 602050
<fseek@GLIBC_2.2.5>
  400726:       68 07 00 00 00          pushq  $0x7
  40072b:       e9 70 ff ff ff          jmpq   4006a0 <.plt>

0000000000400730 <open@plt>:
  400730:       ff 25 22 19 20 00       jmpq   *0x201922(%rip)        # 602058
<open@GLIBC_2.2.5>
  400736:       68 08 00 00 00          pushq  $0x8
  40073b:       e9 60 ff ff ff          jmpq   4006a0 <.plt>

0000000000400740 <fopen@plt>:
  400740:       ff 25 1a 19 20 00       jmpq   *0x20191a(%rip)        # 602060
<fopen@GLIBC_2.2.5>
  400746:       68 09 00 00 00          pushq  $0x9
  40074b:       e9 50 ff ff ff          jmpq   4006a0 <.plt>

0000000000400750 <__isoc99_scanf@plt>:
  400750:       ff 25 12 19 20 00       jmpq   *0x201912(%rip)        # 602068
<__isoc99_scanf@GLIBC_2.7>
  400756:       68 0a 00 00 00          pushq  $0xa
  40075b:       e9 40 ff ff ff          jmpq   4006a0 <.plt>

Disassembly of section .plt.got:

0000000000400760 <printf@plt>:
  400760:	ff 25 82 18 20 00    	jmpq   *0x201882(%rip)        # 601fe8
<printf@GLIBC_2.2.5>
  400766:	66 90                	xchg   %ax,%ax

Disassembly of section .text:

0000000000400770 <_start>:
  400770:	31 ed                	xor    %ebp,%ebp
  400772:	49 89 d1             	mov    %rdx,%r9
  400775:	5e                   pop    %rsi
  400776:	48 89 e2             	mov    %rsp,%rdx
  400779:	48 83 e4 f0          	and    $0xfffffffffffffff0,%rsp
  40077d:	50                   push   %rax
  40077e:	54                   push   %rsp
  40077f:	49 c7 c0 10 0e 40 00 	mov    $0x400e10,%r8
  400786:	48 c7 c1 a0 0d 40 00 	mov    $0x400da0,%rcx
  40078d:	48 c7 c7 57 08 40 00 	mov    $0x400857,%rdi
  400794:	ff 15 56 18 20 00    	callq  *0x201856(%rip)        # 601ff0
<__libc_start_main@GLIBC_2.2.5>
  40079a:	f4                   hlt
  40079b:	0f 1f 44 00 00       	nopl   0x0(%rax,%rax,1)

00000000004007a0 <_dl_relocate_static_pie>:
  4007a0:	f3 c3                	repz retq
  4007a2:	66 2e 0f 1f 84 00 00 	nopw   %cs:0x0(%rax,%rax,1)
  4007a9:	00 00 00
  4007ac:	0f 1f 40 00          	nopl   0x0(%rax)

00000000004007b0 <deregister_tm_clones>:
  4007b0:	55                   push   %rbp
  4007b1:	b8 20 21 60 00       	mov    $0x602120,%eax
  4007b6:	48 3d 20 21 60 00    	cmp    $0x602120,%rax
  4007bc:	48 89 e5             	mov    %rsp,%rbp
  4007bf:	74 17                	je     4007d8 <deregister_tm_clones+0x28>
  4007c1:	b8 00 00 00 00       	mov    $0x0,%eax
  4007c6:	48 85 c0             	test   %rax,%rax
  4007c9:	74 0d                	je     4007d8 <deregister_tm_clones+0x28>
  4007cb:	5d                   pop    %rbp
  4007cc:	bf 20 21 60 00       	mov    $0x602120,%edi
  4007d1:       ff e0                   jmpq   *%rax
  4007d3:       0f 1f 44 00 00          nopl   0x0(%rax,%rax,1)
  4007d8:       5d                      pop    %rbp
  4007d9:       c3                      retq
  4007da:       66 0f 1f 44 00 00       nopw   0x0(%rax,%rax,1)

00000000004007e0 <register_tm_clones>:
  4007e0:       be 20 21 60 00          mov    $0x602120,%esi
  4007e5:       55                      push   %rbp
  4007e6:       48 81 ee 20 21 60 00    sub    $0x602120,%rsi
  4007ed:       48 89 e5                mov    %rsp,%rbp
  4007f0:       48 c1 fe 03             sar    $0x3,%rsi
  4007f4:       48 89 f0                mov    %rsi,%rax
  4007f7:       48 c1 e8 3f             shr    $0x3f,%rax
  4007fb:       48 01 c6                add    %rax,%rsi
  4007fe:       48 d1 fe                sar    %rsi
  400801:       74 15                   je     400818 <register_tm_clones+0x38>
  400803:       b8 00 00 00 00          mov    $0x0,%eax
  400808:       48 85 c0                test   %rax,%rax
  40080b:       74 0b                   je     400818 <register_tm_clones+0x38>
  40080d:       5d                      pop    %rbp
  40080e:       bf 20 21 60 00          mov    $0x602120,%edi
  400813:       ff e0                   jmpq   *%rax
  400815:       0f 1f 00                nopl   (%rax)
  400818:       5d                      pop    %rbp
  400819:       c3                      retq
  40081a:       66 0f 1f 44 00 00       nopw   0x0(%rax,%rax,1)

0000000000400820 <__do_global_dtors_aux>:
  400820:       80 3d f9 18 20 00 00    cmpb   $0x0,0x2018f9(%rip)        # 602120
<__TMC_END__>
  400827:       75 17                   jne    400840 <__do_global_dtors_aux+0x20>
  400829:       55                      push   %rbp
  40082a:       48 89 e5                mov    %rsp,%rbp
  40082d:       e8 7e ff ff ff          callq  4007b0 <deregister_tm_clones>
  400832:       c6 05 e7 18 20 00 01    movb   $0x1,0x2018e7(%rip)        # 602120
<__TMC_END__>
  400839:       5d                      pop    %rbp
  40083a:       c3                      retq
  40083b:       0f 1f 44 00 00          nopl   0x0(%rax,%rax,1)
  400840:       f3 c3                   repz retq
  400842:       0f 1f 40 00             nopl   0x0(%rax)
  400846:       66 2e 0f 1f 84 00 00    nopw   %cs:0x0(%rax,%rax,1)
  40084d:       00 00 00

0000000000400850 <frame_dummy>:
  400850:       55                 push   %rbp
  400851:       48 89 e5                mov    %rsp,%rbp
  400854:       5d                 pop    %rbp
  400855:       eb 89                   jmp    4007e0 <register_tm_clones>

0000000000400857 <main>:
  400857:       55                 push   %rbp
  400858:       48 89 e5                mov    %rsp,%rbp
  40085b:       48 83 ec 10             sub    $0x10,%rsp
  40085f:       64 48 8b 04 25 28 00    mov    %fs:0x28,%rax
  400866:       00 00
  400868:       48 89 45 f8             mov    %rax,-0x8(%rbp)
  40086c:       31 c0                   xor    %eax,%eax
  40086e:       c6 45 f7 61             movb   $0x61,-0x9(%rbp)
  400872:       b8 00 00 00 00          mov    $0x0,%eax
  400877:       e8 75 01 00 00          callq  4009f1 <step1>
  40087c:       eb 24                   jmp    4008a2 <main+0x4b>
  40087e:       48 8d 3d a3 05 00 00    lea    0x5a3(%rip),%rdi        # 400e28
<_IO_stdin_used+0x8>
  400885:       e8 26 fe ff ff          callq  4006b0 <puts@plt>
  40088a:       48 8d 45 f7             lea    -0x9(%rbp),%rax
  40088e:       48 89 c6                mov    %rax,%rsi
  400891:       48 8d 3d a8 05 00 00    lea    0x5a8(%rip),%rdi        # 400e40
<_IO_stdin_used+0x20>
  400898:       b8 00 00 00 00          mov    $0x0,%eax
  40089d:       e8 ae fe ff ff          callq  400750 <__isoc99_scanf@plt>
  4008a2:       0f b6 45 f7             movzbl -0x9(%rbp),%eax
  4008a6:       3c 6e                   cmp    $0x6e,%al
  4008a8:       75 d4                   jne    40087e <main+0x27>
  4008aa:       c6 45 f7 61             movb   $0x61,-0x9(%rbp)
  4008ae:       b8 00 00 00 00          mov    $0x0,%eax
  4008b3:       e8 ba 02 00 00          callq  400b72 <step2>
  4008b8:	eb 24                	jmp    4008de <main+0x87>
  4008ba:	48 8d 3d 83 05 00 00 	lea    0x583(%rip),%rdi        # 400e44
<_IO_stdin_used+0x24>
  4008c1:	e8 ea fd ff ff       	callq  4006b0 <puts@plt>
  4008c6:	48 8d 45 f7          	lea    -0x9(%rbp),%rax
  4008ca:	48 89 c6             	mov    %rax,%rsi
  4008cd:	48 8d 3d 6c 05 00 00 	lea    0x56c(%rip),%rdi        # 400e40
<_IO_stdin_used+0x20>
  4008d4:	b8 00 00 00 00       	mov    $0x0,%eax
  4008d9:	e8 72 fe ff ff       	callq  400750 <__isoc99_scanf@plt>
  4008de:	0f b6 45 f7          	movzbl -0x9(%rbp),%eax
  4008e2:	3c 6e                	cmp    $0x6e,%al
  4008e4:	75 d4                	jne    4008ba <main+0x63>
  4008e6:	c6 45 f7 61          	movb   $0x61,-0x9(%rbp)
  4008ea:	b8 00 00 00 00       	mov    $0x0,%eax
  4008ef:	e8 dd 02 00 00       	callq  400bd1 <step3>
  4008f4:	eb 24                	jmp    40091a <main+0xc3>
  4008f6:	48 8d 3d 5f 05 00 00 	lea    0x55f(%rip),%rdi        # 400e5c
<_IO_stdin_used+0x3c>
  4008fd:	e8 ae fd ff ff       	callq  4006b0 <puts@plt>
  400902:	48 8d 45 f7          	lea    -0x9(%rbp),%rax
  400906:	48 89 c6             	mov    %rax,%rsi
  400909:	48 8d 3d 30 05 00 00 	lea    0x530(%rip),%rdi        # 400e40
<_IO_stdin_used+0x20>
  400910:	b8 00 00 00 00       	mov    $0x0,%eax
  400915:	e8 36 fe ff ff       	callq  400750 <__isoc99_scanf@plt>
  40091a:	0f b6 45 f7          	movzbl -0x9(%rbp),%eax
  40091e:	3c 6e                	cmp    $0x6e,%al
  400920:	75 d4                	jne    4008f6 <main+0x9f>
  400922:	c6 45 f7 61          	movb   $0x61,-0x9(%rbp)
  400926:	b8 00 00 00 00       	mov    $0x0,%eax
  40092b:	e8 b7 02 00 00       	callq  400be7 <step4>
  400930:	eb 24                	jmp    400956 <main+0xff>
  400932:	48 8d 3d 3b 05 00 00 	lea    0x53b(%rip),%rdi        # 400e74
<_IO_stdin_used+0x54>
  400939:	e8 72 fd ff ff       	callq  4006b0 <puts@plt>
  40093e:	48 8d 45 f7          	lea    -0x9(%rbp),%rax
  400942:	48 89 c6             	mov    %rax,%rsi
  400945:	48 8d 3d f4 04 00 00 	lea    0x4f4(%rip),%rdi        # 400e40 
<_IO_stdin_used+0x20>
  40094c:	b8 00 00 00 00       	mov    $0x0,%eax
  400951:	e8 fa fd ff ff       	callq  400750 <__isoc99_scanf@plt>
  400956:	0f b6 45 f7          	movzbl -0x9(%rbp),%eax
  40095a:	3c 6e                	cmp    $0x6e,%al
  40095c:	75 d4                	jne    400932 <main+0xdb>
  40095e:	c6 45 f7 61          	movb   $0x61,-0x9(%rbp)
  400962:	b8 00 00 00 00       	mov    $0x0,%eax
  400967:	e8 30 03 00 00       	callq  400c9c <step5>
  40096c:	eb 24                	jmp    400992 <main+0x13b>
  40096e:	48 8d 3d 17 05 00 00 	lea    0x517(%rip),%rdi        # 400e8c 
<_IO_stdin_used+0x6c>
  400975:	e8 36 fd ff ff       	callq  4006b0 <puts@plt>
  40097a:	48 8d 45 f7          	lea    -0x9(%rbp),%rax
  40097e:	48 89 c6             	mov    %rax,%rsi
  400981:	48 8d 3d b8 04 00 00 	lea    0x4b8(%rip),%rdi        # 400e40 
<_IO_stdin_used+0x20>
  400988:	b8 00 00 00 00       	mov    $0x0,%eax
  40098d:	e8 be fd ff ff       	callq  400750 <__isoc99_scanf@plt>
  400992:	0f b6 45 f7          	movzbl -0x9(%rbp),%eax
  400996:	3c 6e                	cmp    $0x6e,%al
  400998:	75 d4                	jne    40096e <main+0x117>
  40099a:	c6 45 f7 61          	movb   $0x61,-0x9(%rbp)
  40099e:	b8 00 00 00 00       	mov    $0x0,%eax
  4009a3:	e8 23 03 00 00       	callq  400ccb <step6>
  4009a8:	eb 24                	jmp    4009ce <main+0x177>
  4009aa:	48 8d 3d f3 04 00 00 	lea    0x4f3(%rip),%rdi        # 400ea4 
<_IO_stdin_used+0x84>
  4009b1:	e8 fa fc ff ff       	callq  4006b0 <puts@plt>
  4009b6:	48 8d 45 f7          	lea    -0x9(%rbp),%rax
  4009ba:	48 89 c6             	mov    %rax,%rsi
  4009bd:	48 8d 3d 7c 04 00 00 	lea    0x47c(%rip),%rdi        # 400e40 
<_IO_stdin_used+0x20>
  4009c4:	b8 00 00 00 00       	mov    $0x0,%eax
  4009c9:	e8 82 fd ff ff       	callq  400750 <__isoc99_scanf@plt>
  4009ce:	0f b6 45 f7          	movzbl -0x9(%rbp),%eax
  4009d2:	3c 6e                	cmp    $0x6e,%al
  4009d4:	75 d4                	jne    4009aa <main+0x153>
  4009d6:       b8 00 00 00 00          mov    $0x0,%eax
  4009db:       48 8b 55 f8             mov    -0x8(%rbp),%rdx
  4009df:       64 48 33 14 25 28 00  xor    %fs:0x28,%rdx
  4009e6:       00 00
  4009e8:       74 05                   je     4009ef <main+0x198>
  4009ea:       e8 e1 fc ff ff          callq  4006d0 <__stack_chk_fail@plt>
  4009ef:       c9                   leaveq
  4009f0:       c3                   retq

00000000004009f1 <step1>:
  4009f1:       55                   push   %rbp
  4009f2:       48 89 e5                mov    %rsp,%rbp
  4009f5:       48 8d 05 a4 16 20 00  lea    0x2016a4(%rip),%rax        # 6020a0 <initialized>
  4009fc:       48 89 c6                mov    %rax,%rsi
  4009ff:       48 8d 3d b5 04 00 00  lea    0x4b5(%rip),%rdi        # 400ebb
<_IO_stdin_used+0x9b>
  400a06:       b8 00 00 00 00          mov    $0x0,%eax
  400a0b:       e8 50 fd ff ff          callq  400760 <printf@plt>
  400a10:       48 8d 05 49 17 20 00  lea    0x201749(%rip),%rax        # 602160 <un_initialized>
  400a17:       48 89 c6                mov    %rax,%rsi
  400a1a:       48 8d 3d b7 04 00 00  lea    0x4b7(%rip),%rdi        # 400ed8
<_IO_stdin_used+0xb8>
  400a21:       b8 00 00 00 00          mov    $0x0,%eax
  400a26:       e8 35 fd ff ff          callq  400760 <printf@plt>
  400a2b:       48 8d 05 0e 17 20 00  lea    0x20170e(%rip),%rax        # 602140 <un_init_ptr1>
  400a32:       48 89 c6                mov    %rax,%rsi
  400a35:       48 8d 3d bc 04 00 00  lea    0x4bc(%rip),%rdi        # 400ef8
<_IO_stdin_used+0xd8>
  400a3c:       b8 00 00 00 00          mov    $0x0,%eax
  400a41:       e8 1a fd ff ff          callq  400760 <printf@plt>
  400a46:       48 8d 05 fb 16 20 00  lea    0x2016fb(%rip),%rax        # 602148 <un_init_ptr2>
  400a4d:       48 89 c6                mov    %rax,%rsi
  400a50:       48 8d 3d bf 04 00 00  lea    0x4bf(%rip),%rdi        # 400f16
<_IO_stdin_used+0xf6>
  400a57:       b8 00 00 00 00          mov    $0x0,%eax
  400a5c:       e8 ff fc ff ff          callq  400760 <printf@plt>
  400a61:       48 8d 05 10 06 00 00  lea    0x610(%rip),%rax        # 401078 <init_const>
  400a68:       48 89 c6                mov    %rax,%rsi
  400a6b:	48 8d 3d c2 04 00 00 	lea    0x4c2(%rip),%rdi        # 400f34
<_IO_stdin_used+0x114>
  400a72:	b8 00 00 00 00       	mov    $0x0,%eax
  400a77:	e8 e4 fc ff ff       	callq  400760 <printf@plt>
  400a7c:	48 8d 05 d4 fd ff ff 	lea    -0x22c(%rip),%rax        # 400857 <main>
  400a83:	48 89 c6             	mov    %rax,%rsi
  400a86:	48 8d 3d c0 04 00 00 	lea    0x4c0(%rip),%rdi        # 400f4d
<_IO_stdin_used+0x12d>
  400a8d:	b8 00 00 00 00       	mov    $0x0,%eax
  400a92:	e8 c9 fc ff ff       	callq  400760 <printf@plt>
  400a97:	48 8d 05 53 ff ff ff 	lea    -0xad(%rip),%rax        # 4009f1 <step1>
  400a9e:	48 89 c6             	mov    %rax,%rsi
  400aa1:	48 8d 3d b7 04 00 00 	lea    0x4b7(%rip),%rdi        # 400f5f
<_IO_stdin_used+0x13f>
  400aa8:	b8 00 00 00 00       	mov    $0x0,%eax
  400aad:	e8 ae fc ff ff       	callq  400760 <printf@plt>
  400ab2:	48 8d 05 b9 00 00 00 	lea    0xb9(%rip),%rax        # 400b72 <step2>
  400ab9:	48 89 c6             	mov    %rax,%rsi
  400abc:	48 8d 3d af 04 00 00 	lea    0x4af(%rip),%rdi        # 400f72
<_IO_stdin_used+0x152>
  400ac3:	b8 00 00 00 00       	mov    $0x0,%eax
  400ac8:	e8 93 fc ff ff       	callq  400760 <printf@plt>
  400acd:	48 8d 05 fd 00 00 00 	lea    0xfd(%rip),%rax        # 400bd1 <step3>
  400ad4:	48 89 c6             	mov    %rax,%rsi
  400ad7:	48 8d 3d a7 04 00 00 	lea    0x4a7(%rip),%rdi        # 400f85
<_IO_stdin_used+0x165>
  400ade:	b8 00 00 00 00       	mov    $0x0,%eax
  400ae3:	e8 78 fc ff ff       	callq  400760 <printf@plt>
  400ae8:	48 8d 05 f8 00 00 00 	lea    0xf8(%rip),%rax        # 400be7 <step4>
  400aef:	48 89 c6             	mov    %rax,%rsi
  400af2:	48 8d 3d 9f 04 00 00 	lea    0x49f(%rip),%rdi        # 400f98
<_IO_stdin_used+0x178>
  400af9:	b8 00 00 00 00       	mov    $0x0,%eax
  400afe:	e8 5d fc ff ff       	callq  400760 <printf@plt>
  400b03:	48 8d 05 92 01 00 00 	lea    0x192(%rip),%rax        # 400c9c <step5>
  400b0a:	48 89 c6             	mov    %rax,%rsi
  400b0d:	48 8d 3d 97 04 00 00 	lea    0x497(%rip),%rdi        # 400fab
<_IO_stdin_used+0x18b>
  400b14:	b8 00 00 00 00       	mov    $0x0,%eax
  400b19:       e8 42 fc ff ff          callq  400760 <printf@plt>
  400b1e:       48 8d 05 a6 01 00 00    lea    0x1a6(%rip),%rax        # 400ccb <step6>
  400b25:       48 89 c6                mov    %rax,%rsi
  400b28:       48 8d 3d 8f 04 00 00    lea    0x48f(%rip),%rdi        # 400fbe
<_IO_stdin_used+0x19e>
  400b2f:       b8 00 00 00 00          mov    $0x0,%eax
  400b34:       e8 27 fc ff ff          callq  400760 <printf@plt>
  400b39:       48 c7 c0 3d 0d 40 00    mov    $0x400d3d,%rax
  400b40:       48 89 c6                mov    %rax,%rsi
  400b43:       48 8d 3d 87 04 00 00    lea    0x487(%rip),%rdi        # 400fd1
<_IO_stdin_used+0x1b1>
  400b4a:       b8 00 00 00 00          mov    $0x0,%eax
  400b4f:       e8 0c fc ff ff          callq  400760 <printf@plt>
  400b54:       48 8b 05 8d 14 20 00    mov    0x20148d(%rip),%rax        # 601fe8
<printf@GLIBC_2.2.5>
  400b5b:       48 89 c6                mov    %rax,%rsi
  400b5e:       48 8d 3d 7d 04 00 00    lea    0x47d(%rip),%rdi        # 400fe2
<_IO_stdin_used+0x1c2>
  400b65:       b8 00 00 00 00          mov    $0x0,%eax
  400b6a:       e8 f1 fb ff ff          callq  400760 <printf@plt>
  400b6f:       90                      nop
  400b70:       5d                      pop    %rbp
  400b71:       c3                      retq

0000000000400b72 <step2>:
  400b72:       55                      push   %rbp
  400b73:       48 89 e5                mov    %rsp,%rbp
  400b76:       bf 80 84 1e 00          mov    $0x1e8480,%edi
  400b7b:       e8 90 fb ff ff          callq  400710 <malloc@plt>
  400b80:       48 89 05 b9 15 20 00    mov    %rax,0x2015b9(%rip)        # 602140 <un_init_ptr1>
  400b87:       bf 80 84 1e 00          mov    $0x1e8480,%edi
  400b8c:       e8 7f fb ff ff          callq  400710 <malloc@plt>
  400b91:       48 89 05 b0 15 20 00    mov    %rax,0x2015b0(%rip)        # 602148 <un_init_ptr2>
  400b98:       48 8b 05 a1 15 20 00    mov    0x2015a1(%rip),%rax        # 602140 <un_init_ptr1>
  400b9f:       48 89 c6                mov    %rax,%rsi
  400ba2:       48 8d 3d 4f 04 00 00    lea    0x44f(%rip),%rdi        # 400ff8
<_IO_stdin_used+0x1d8>
  400ba9:       b8 00 00 00 00          mov    $0x0,%eax
  400bae:       e8 ad fb ff ff          callq  400760 <printf@plt>
  400bb3:        48 8b 05 8e 15 20 00  mov    0x20158e(%rip),%rax        # 602148 <un_init_ptr2>
  400bba:        48 89 c6              mov    %rax,%rsi
  400bbd:        48 8d 3d 5c 04 00 00  lea    0x45c(%rip),%rdi        # 401020
<_IO_stdin_used+0x200>
  400bc4:        b8 00 00 00 00        mov    $0x0,%eax
  400bc9:        e8 92 fb ff ff        callq  400760 <printf@plt>
  400bce:        90                nop
  400bcf:        5d                pop    %rbp
  400bd0:        c3                retq

0000000000400bd1 <step3>:
  400bd1:        55                push   %rbp
  400bd2:        48 89 e5              mov    %rsp,%rbp
  400bd5:        bf d4 30 00 00        mov    $0x30d4,%edi
  400bda:        b8 00 00 00 00        mov    $0x0,%eax
  400bdf:        e8 59 01 00 00        callq  400d3d <foo>
  400be4:        90                nop
  400be5:        5d                pop    %rbp
  400be6:        c3                retq

0000000000400be7 <step4>:
  400be7:        55                push   %rbp
  400be8:        48 89 e5              mov    %rsp,%rbp
  400beb:        48 83 ec 10           sub    $0x10,%rsp
  400bef:        48 8d 35 4c 04 00 00  lea    0x44c(%rip),%rsi        # 401042
<_IO_stdin_used+0x222>
  400bf6:        48 8d 3d 47 04 00 00  lea    0x447(%rip),%rdi        # 401044
<_IO_stdin_used+0x224>
  400bfd:        e8 3e fb ff ff        callq  400740 <fopen@plt>
  400c02:        48 89 45 f8           mov    %rax,-0x8(%rbp)
  400c06:        48 8b 45 f8           mov    -0x8(%rbp),%rax
  400c0a:        ba 00 00 00 00        mov    $0x0,%edx
  400c0f:        be 80 84 1e 00        mov    $0x1e8480,%esi
  400c14:        48 89 c7              mov    %rax,%rdi
  400c17:        e8 04 fb ff ff        callq  400720 <fseek@plt>
  400c1c:        48 8b 45 f8           mov    -0x8(%rbp),%rax
  400c20:        48 89 c6              mov    %rax,%rsi
  400c23:        bf 00 00 00 00        mov    $0x0,%edi
  400c28:        e8 d3 fa ff ff        callq  400700 <fputc@plt>
  400c2d:       48 8b 45 f8             mov    -0x8(%rbp),%rax
  400c31:       48 89 c7                mov    %rax,%rdi
  400c34:       e8 87 fa ff ff          callq  4006c0 <fclose@plt>
  400c39:       be 02 00 00 00          mov    $0x2,%esi
  400c3e:       48 8d 3d ff 03 00 00    lea    0x3ff(%rip),%rdi        # 401044
<_IO_stdin_used+0x224>
  400c45:       b8 00 00 00 00          mov    $0x0,%eax
  400c4a:       e8 e1 fa ff ff          callq  400730 <open@plt>
  400c4f:       89 45 f4                mov    %eax,-0xc(%rbp)
  400c52:       8b 45 f4                mov    -0xc(%rbp),%eax
  400c55:       41 b9 00 00 00 00       mov    $0x0,%r9d
  400c5b:       41 89 c0                mov    %eax,%r8d
  400c5e:       b9 02 00 00 00          mov    $0x2,%ecx
  400c63:       ba 03 00 00 00          mov    $0x3,%edx
  400c68:       be 80 84 1e 00          mov    $0x1e8480,%esi
  400c6d:       bf 00 00 00 00          mov    $0x0,%edi
  400c72:       e8 69 fa ff ff          callq  4006e0 <mmap@plt>
  400c77:       48 89 05 aa 14 20 00    mov    %rax,0x2014aa(%rip)        # 602128 <region>
  400c7e:       48 8b 05 a3 14 20 00    mov    0x2014a3(%rip),%rax        # 602128 <region>
  400c85:       48 89 c6                mov    %rax,%rsi
  400c88:       48 8d 3d bd 03 00 00    lea    0x3bd(%rip),%rdi        # 40104c
<_IO_stdin_used+0x22c>
  400c8f:       b8 00 00 00 00          mov    $0x0,%eax
  400c94:       e8 c7 fa ff ff          callq  400760 <printf@plt>
  400c99:       90                      nop
  400c9a:       c9                      leaveq
  400c9b:       c3                      retq

0000000000400c9c <step5>:
  400c9c:       55                      push   %rbp
  400c9d:       48 89 e5                mov    %rsp,%rbp
  400ca0:       c7 45 fc 00 00 00 00    movl   $0x0,-0x4(%rbp)
  400ca7:       eb 16                   jmp    400cbf <step5+0x23>
  400ca9:       48 8b 15 78 14 20 00    mov    0x201478(%rip),%rdx        # 602128 <region>
  400cb0:       8b 45 fc                mov    -0x4(%rbp),%eax
  400cb3:       48 98                   cltq
  400cb5:       48 01 d0                add    %rdx,%rax
  400cb8:       c6 00 6c                movb   $0x6c,(%rax)
  400cbb:       83 45 fc 01             addl   $0x1,-0x4(%rbp)
  400cbf:	81 7d fc 6f 17 00 00 	cmpl   $0x176f,-0x4(%rbp)
  400cc6:	7e e1                	jle    400ca9 <step5+0xd>
  400cc8:	90                   nop
  400cc9:	5d                   pop    %rbp
  400cca:	c3                   retq

0000000000400ccb <step6>:
  400ccb:	55                   push   %rbp
  400ccc:	48 89 e5             	mov    %rsp,%rbp
  400ccf:	48 83 ec 10          	sub    $0x10,%rsp
  400cd3:	48 c7 45 f8 00 00 40 	movq   $0x400000,-0x8(%rbp)
  400cda:	00
  400cdb:	c7 45 f4 00 00 00 00 	movl   $0x0,-0xc(%rbp)
  400ce2:	eb 4d                	jmp    400d31 <step6+0x66>
  400ce4:	8b 45 f4             	mov    -0xc(%rbp),%eax
  400ce7:	48 98                	cltq
  400ce9:	48 8d 14 85 00 00 00 	lea    0x0(,%rax,4),%rdx
  400cf0:	00
  400cf1:	48 8b 45 f8          	mov    -0x8(%rbp),%rax
  400cf5:	48 01 d0             	add    %rdx,%rax
  400cf8:	8b 00                	mov    (%rax),%eax
  400cfa:	89 c7                	mov    %eax,%edi
  400cfc:	e8 ef f9 ff ff       	callq  4006f0 <htonl@plt>
  400d01:	89 c1                	mov    %eax,%ecx
  400d03:	8b 45 f4             	mov    -0xc(%rbp),%eax
  400d06:	48 98                	cltq
  400d08:	48 8d 14 85 00 00 00 	lea    0x0(,%rax,4),%rdx
  400d0f:	00
  400d10:	48 8b 45 f8          	mov    -0x8(%rbp),%rax
  400d14:	48 01 d0             	add    %rdx,%rax
  400d17:	89 ca                	mov    %ecx,%edx
  400d19:	48 89 c6             	mov    %rax,%rsi
  400d1c:	48 8d 3d 46 03 00 00 	lea    0x346(%rip),%rdi        # 401069
<_IO_stdin_used+0x249>
  400d23:	b8 00 00 00 00       	mov    $0x0,%eax
  400d28:	e8 33 fa ff ff       	callq  400760 <printf@plt>
  400d2d:	83 45 f4 01          	addl   $0x1,-0xc(%rbp)
  400d31:	81 7d f4 ff 03 00 00 	cmpl   $0x3ff,-0xc(%rbp)
  400d38:	7e aa                	jle    400ce4 <step6+0x19>
  400d3a:       90                   nop
  400d3b:       c9                   leaveq
  400d3c:       c3                   retq

0000000000400d3d <foo>:
  400d3d:       55                   push   %rbp
  400d3e:       48 89 e5                     mov    %rsp,%rbp
  400d41:       48 83 ec 20                  sub    $0x20,%rsp
  400d45:       89 7d ec                     mov    %edi,-0x14(%rbp)
  400d48:       8b 45 ec                     mov    -0x14(%rbp),%eax
  400d4b:       85 c0                        test   %eax,%eax
  400d4d:       7e 40                        jle    400d8f <foo+0x52>
  400d4f:       48 c7 45 f8 01 00 00   movq   $0x1,-0x8(%rbp)
  400d56:       00
  400d57:       8b 45 ec                     mov    -0x14(%rbp),%eax
  400d5a:       3d d4 30 00 00               cmp    $0x30d4,%eax
  400d5f:       74 08                        je     400d69 <foo+0x2c>
  400d61:       8b 45 ec                     mov    -0x14(%rbp),%eax
  400d64:       83 f8 01                     cmp    $0x1,%eax
  400d67:       75 18                        jne    400d81 <foo+0x44>
  400d69:       48 8d 45 ec                  lea    -0x14(%rbp),%rax
  400d6d:       48 89 c6                     mov    %rax,%rsi
  400d70:       48 8d 3d 09 03 00 00  lea    0x309(%rip),%rdi        # 401080 <init_const+0x8>
  400d77:       b8 00 00 00 00               mov    $0x0,%eax
  400d7c:       e8 df f9 ff ff               callq  400760 <printf@plt>
  400d81:       8b 45 ec                     mov    -0x14(%rbp),%eax
  400d84:       83 e8 01                     sub    $0x1,%eax
  400d87:       89 c7                        mov    %eax,%edi
  400d89:       e8 af ff ff ff               callq  400d3d <foo>
  400d8e:       90                   nop
  400d8f:       90                   nop
  400d90:       c9                   leaveq
  400d91:       c3                   retq
  400d92:       66 2e 0f 1f 84 00 00   nopw   %cs:0x0(%rax,%rax,1)
  400d99:       00 00 00
  400d9c:       0f 1f 40 00                  nopl   0x0(%rax)

0000000000400da0 <__libc_csu_init>:
  400da0:       41 57                        push   %r15
  400da2:      41 56                   push   %r14
  400da4:      49 89 d7                mov    %rdx,%r15
  400da7:      41 55                   push   %r13
  400da9:      41 54                   push   %r12
  400dab:      4c 8d 25 56 10 20 00  lea    0x201056(%rip),%r12        # 601e08
<__frame_dummy_init_array_entry>
  400db2:      55                 push   %rbp
  400db3:      48 8d 2d 56 10 20 00  lea    0x201056(%rip),%rbp        # 601e10
<__init_array_end>
  400dba:      53                 push   %rbx
  400dbb:      41 89 fd                mov    %edi,%r13d
  400dbe:      49 89 f6                mov    %rsi,%r14
  400dc1:      4c 29 e5                sub    %r12,%rbp
  400dc4:      48 83 ec 08             sub    $0x8,%rsp
  400dc8:      48 c1 fd 03             sar    $0x3,%rbp
  400dcc:      e8 af f8 ff ff          callq  400680 <_init>
  400dd1:      48 85 ed                test   %rbp,%rbp
  400dd4:      74 20                   je     400df6 <__libc_csu_init+0x56>
  400dd6:      31 db                   xor    %ebx,%ebx
  400dd8:      0f 1f 84 00 00 00 00   nopl   0x0(%rax,%rax,1)
  400ddf:      00
  400de0:      4c 89 fa                mov    %r15,%rdx
  400de3:      4c 89 f6                mov    %r14,%rsi
  400de6:      44 89 ef                mov    %r13d,%edi
  400de9:      41 ff 14 dc             callq  *(%r12,%rbx,8)
  400ded:      48 83 c3 01             add    $0x1,%rbx
  400df1:      48 39 dd                cmp    %rbx,%rbp
  400df4:      75 ea                   jne    400de0 <__libc_csu_init+0x40>
  400df6:      48 83 c4 08             add    $0x8,%rsp
  400dfa:      5b                 pop    %rbx
  400dfb:      5d                 pop    %rbp
  400dfc:      41 5c                   pop    %r12
  400dfe:      41 5d                   pop    %r13
  400e00:      41 5e                   pop    %r14
  400e02:      41 5f                   pop    %r15
  400e04:      c3                 retq
  400e05:      90                 nop
  400e06:      66 2e 0f 1f 84 00 00   nopw   %cs:0x0(%rax,%rax,1)
  400e0d:      00 00 00
```

```
0000000000400e10 <__libc_csu_fini>:
  400e10:       f3 c3                   repz retq

Disassembly of section .fini:

0000000000400e14 <_fini>:
  400e14:       48 83 ec 08             sub    $0x8,%rsp
  400e18:       48 83 c4 08             add    $0x8,%rsp
```

OBJDUMP of module1.o:

```
module1.o:     file format elf64-x86-64
module1.o
architecture: i386:x86-64, flags 0x00000011:
HAS_RELOC, HAS_SYMS
start address 0x0000000000000000

Sections:
Idx Name          Size      VMA               LMA               File off  Algn
  0 .text         000004e6  0000000000000000  0000000000000000  00000040  2**0
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  0000000000000000  0000000000000000  00000526  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  0000000000000000  0000000000000000  00000526  2**0
                  ALLOC
  3 .rodata       0000024e  0000000000000000  0000000000000000  00000528  2**3
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .comment      0000002a  0000000000000000  0000000000000000  00000776  2**0
                  CONTENTS, READONLY
  5 .note.GNU-stack 00000000  0000000000000000  0000000000000000  000007a0  2**0
                  CONTENTS, READONLY
  6 .eh_frame     000000f8  0000000000000000  0000000000000000  000007a0  2**3
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, DATA
SYMBOL TABLE:
0000000000000000 l    df *ABS*  0000000000000000 module1.c
0000000000000000 l    d  .text  0000000000000000 .text
0000000000000000 l    d  .data  0000000000000000 .data
0000000000000000 l    d  .bss   0000000000000000 .bss
0000000000000000 l    d  .rodata        0000000000000000 .rodata
```

```
0000000000000000 l    d  .note.GNU-stack   0000000000000000 .note.GNU-stack
0000000000000000 l    d  .eh_frame  0000000000000000 .eh_frame
0000000000000000 l    d  .comment  0000000000000000 .comment
0000000000000008       O *COM*    0000000000000008 region
0000000000000000 g     F .text         000000000000019a main
000000000000019a g     F .text         0000000000000181 step1
0000000000000000         *UND*      0000000000000000 _GLOBAL_OFFSET_TABLE_
0000000000000000         *UND*      0000000000000000 puts
0000000000000000         *UND*      0000000000000000 __isoc99_scanf
000000000000031b g     F .text         000000000000005f step2
000000000000037a g     F .text         0000000000000016 step3
0000000000000390 g     F .text         00000000000000b5 step4
0000000000000445 g     F .text         000000000000002f step5
0000000000000474 g     F .text         0000000000000072 step6
0000000000000000         *UND*      0000000000000000 __stack_chk_fail
0000000000000000         *UND*      0000000000000000 initialized
0000000000000000         *UND*      0000000000000000 printf
0000000000000000         *UND*      0000000000000000 un_initialized
0000000000000000         *UND*      0000000000000000 un_init_ptr1
0000000000000000         *UND*      0000000000000000 un_init_ptr2
0000000000000000         *UND*      0000000000000000 init_const
0000000000000000         *UND*      0000000000000000 foo
0000000000000000         *UND*      0000000000000000 malloc
0000000000000000         *UND*      0000000000000000 fopen
0000000000000000         *UND*      0000000000000000 fseek
0000000000000000         *UND*      0000000000000000 fputc
0000000000000000         *UND*      0000000000000000 fclose
0000000000000000         *UND*      0000000000000000 open
0000000000000000         *UND*      0000000000000000 mmap
0000000000000000         *UND*      0000000000000000 htonl



Disassembly of section .text:

0000000000000000 <main>:
   0:	55                   	push   %rbp
   1:	48 89 e5             	mov    %rsp,%rbp
   4:	48 83 ec 10          	sub    $0x10,%rsp
```

```
 8:   64 48 8b 04 25 28 00  mov    %fs:0x28,%rax
 f:   00 00
11:   48 89 45 f8           mov    %rax,-0x8(%rbp)
15:   31 c0                 xor    %eax,%eax
17:   c6 45 f7 61           movb   $0x61,-0x9(%rbp)
1b:   b8 00 00 00 00        mov    $0x0,%eax
20:   e8 00 00 00 00        callq  25 <main+0x25>
                    21: R_X86_64_PC32  step1-0x4
25:   eb 24                 jmp    4b <main+0x4b>
27:   48 8d 3d 00 00 00 00  lea    0x0(%rip),%rdi        # 2e <main+0x2e>
                    2a: R_X86_64_PC32  .rodata-0x4
2e:   e8 00 00 00 00        callq  33 <main+0x33>
                    2f: R_X86_64_PLT32puts-0x4
33:   48 8d 45 f7           lea    -0x9(%rbp),%rax
37:   48 89 c6              mov    %rax,%rsi
3a:   48 8d 3d 00 00 00 00  lea    0x0(%rip),%rdi        # 41 <main+0x41>
                    3d: R_X86_64_PC32  .rodata+0x14
41:   b8 00 00 00 00        mov    $0x0,%eax
46:   e8 00 00 00 00        callq  4b <main+0x4b>
                    47: R_X86_64_PLT32        __isoc99_scanf-0x4
4b:   0f b6 45 f7           movzbl -0x9(%rbp),%eax
4f:   3c 6e                 cmp    $0x6e,%al
51:   75 d4                 jne    27 <main+0x27>
53:   c6 45 f7 61           movb   $0x61,-0x9(%rbp)
57:   b8 00 00 00 00        mov    $0x0,%eax
5c:   e8 00 00 00 00        callq  61 <main+0x61>
                    5d: R_X86_64_PC32  step2-0x4
61:   eb 24                 jmp    87 <main+0x87>
63:   48 8d 3d 00 00 00 00  lea    0x0(%rip),%rdi        # 6a <main+0x6a>
                    66: R_X86_64_PC32  .rodata+0x18
6a:   e8 00 00 00 00        callq  6f <main+0x6f>
                    6b: R_X86_64_PLT32        puts-0x4
6f:   48 8d 45 f7           lea    -0x9(%rbp),%rax
73:   48 89 c6              mov    %rax,%rsi
76:   48 8d 3d 00 00 00 00  lea    0x0(%rip),%rdi        # 7d <main+0x7d>
                    79: R_X86_64_PC32  .rodata+0x14
7d:   b8 00 00 00 00        mov    $0x0,%eax
82:   e8 00 00 00 00        callq  87 <main+0x87>
                    83: R_X86_64_PLT32        __isoc99_scanf-0x4
```

```
87:    0f b6 45 f7             movzbl -0x9(%rbp),%eax
8b:    3c 6e                   cmp    $0x6e,%al
8d:    75 d4                   jne    63 <main+0x63>
8f:    c6 45 f7 61             movb   $0x61,-0x9(%rbp)
93:    b8 00 00 00 00          mov    $0x0,%eax
98:    e8 00 00 00 00          callq  9d <main+0x9d>
                       99: R_X86_64_PC32  step3-0x4
9d:    eb 24                   jmp    c3 <main+0xc3>
9f:    48 8d 3d 00 00 00 00    lea    0x0(%rip),%rdi        # a6 <main+0xa6>
                       a2: R_X86_64_PC32  .rodata+0x30
a6:    e8 00 00 00 00          callq  ab <main+0xab>
                       a7: R_X86_64_PLT32       puts-0x4
ab:    48 8d 45 f7             lea    -0x9(%rbp),%rax
af:    48 89 c6                mov    %rax,%rsi
b2:    48 8d 3d 00 00 00 00    lea    0x0(%rip),%rdi        # b9 <main+0xb9>
                       b5: R_X86_64_PC32  .rodata+0x14
b9:    b8 00 00 00 00          mov    $0x0,%eax
be:    e8 00 00 00 00          callq  c3 <main+0xc3>
                       bf: R_X86_64_PLT32__isoc99_scanf-0x4
c3:    0f b6 45 f7             movzbl -0x9(%rbp),%eax
c7:    3c 6e                   cmp    $0x6e,%al
c9:    75 d4                   jne    9f <main+0x9f>
cb:    c6 45 f7 61             movb   $0x61,-0x9(%rbp)
cf:    b8 00 00 00 00          mov    $0x0,%eax
d4:    e8 00 00 00 00          callq  d9 <main+0xd9>
                       d5: R_X86_64_PC32  step4-0x4
d9:    eb 24                   jmp    ff <main+0xff>
db:    48 8d 3d 00 00 00 00    lea    0x0(%rip),%rdi        # e2 <main+0xe2>
                       de: R_X86_64_PC32  .rodata+0x48
e2:    e8 00 00 00 00          callq  e7 <main+0xe7>
                       e3: R_X86_64_PLT32       puts-0x4
e7:    48 8d 45 f7             lea    -0x9(%rbp),%rax
eb:    48 89 c6                mov    %rax,%rsi
ee:    48 8d 3d 00 00 00 00    lea    0x0(%rip),%rdi        # f5 <main+0xf5>
                       f1: R_X86_64_PC32  .rodata+0x14
f5:    b8 00 00 00 00          mov    $0x0,%eax
fa:    e8 00 00 00 00          callq  ff <main+0xff>
                       fb: R_X86_64_PLT32__isoc99_scanf-0x4
ff:    0f b6 45 f7             movzbl -0x9(%rbp),%eax
```

```
103:   3c 6e                 cmp    $0x6e,%al
105:   75 d4                 jne    db <main+0xdb>
107:   c6 45 f7 61           movb   $0x61,-0x9(%rbp)
10b:   b8 00 00 00 00        mov    $0x0,%eax
110:   e8 00 00 00 00        callq  115 <main+0x115>
                      111: R_X86_64_PC32       step5-0x4
115:   eb 24                 jmp    13b <main+0x13b>
117:   48 8d 3d 00 00 00 00  lea    0x0(%rip),%rdi       # 11e <main+0x11e>
                      11a: R_X86_64_PC32       .rodata+0x60
11e:   e8 00 00 00 00        callq  123 <main+0x123>
                      11f: R_X86_64_PLT32      puts-0x4
123:   48 8d 45 f7           lea    -0x9(%rbp),%rax
127:   48 89 c6              mov    %rax,%rsi
12a:   48 8d 3d 00 00 00 00  lea    0x0(%rip),%rdi       # 131 <main+0x131>
                      12d: R_X86_64_PC32       .rodata+0x14
131:   b8 00 00 00 00        mov    $0x0,%eax
136:   e8 00 00 00 00        callq  13b <main+0x13b>
                      137: R_X86_64_PLT32      __isoc99_scanf-0x4
13b:   0f b6 45 f7           movzbl -0x9(%rbp),%eax
13f:   3c 6e                 cmp    $0x6e,%al
141:   75 d4                 jne    117 <main+0x117>
143:   c6 45 f7 61           movb   $0x61,-0x9(%rbp)
147:   b8 00 00 00 00        mov    $0x0,%eax
14c:   e8 00 00 00 00        callq  151 <main+0x151>
                      14d: R_X86_64_PC32       step6-0x4
151:   eb 24                 jmp    177 <main+0x177>
153:   48 8d 3d 00 00 00 00  lea    0x0(%rip),%rdi       # 15a <main+0x15a>
                      156: R_X86_64_PC32       .rodata+0x78
15a:   e8 00 00 00 00        callq  15f <main+0x15f>
                      15b: R_X86_64_PLT32      puts-0x4
15f:   48 8d 45 f7           lea    -0x9(%rbp),%rax
163:   48 89 c6              mov    %rax,%rsi
166:   48 8d 3d 00 00 00 00  lea    0x0(%rip),%rdi       # 16d <main+0x16d>
                      169: R_X86_64_PC32       .rodata+0x14
16d:   b8 00 00 00 00        mov    $0x0,%eax
172:   e8 00 00 00 00        callq  177 <main+0x177>
                      173: R_X86_64_PLT32      __isoc99_scanf-0x4
177:   0f b6 45 f7           movzbl -0x9(%rbp),%eax
17b:   3c 6e                 cmp    $0x6e,%al
```

```
 17d:   75 d4                   jne    153 <main+0x153>
 17f:   b8 00 00 00 00          mov    $0x0,%eax
 184:   48 8b 55 f8             mov    -0x8(%rbp),%rdx
 188:   64 48 33 14 25 28 00    xor    %fs:0x28,%rdx
 18f:   00 00
 191:   74 05                   je     198 <main+0x198>
 193:   e8 00 00 00 00          callq  198 <main+0x198>
                        194: R_X86_64_PLT32     __stack_chk_fail-0x4
 198:   c9                      leaveq
 199:   c3                      retq

000000000000019a <step1>:
 19a:   55                      push   %rbp
 19b:   48 89 e5                mov    %rsp,%rbp
 19e:   48 8d 05 00 00 00 00    lea    0x0(%rip),%rax        # 1a5 <step1+0xb>
                        1a1: R_X86_64_PC32      initialized-0x4
 1a5:   48 89 c6                mov    %rax,%rsi
 1a8:   48 8d 3d 00 00 00 00    lea    0x0(%rip),%rdi        # 1af <step1+0x15>
                        1ab: R_X86_64_PC32      .rodata+0x8f
 1af:   b8 00 00 00 00          mov    $0x0,%eax
 1b4:   e8 00 00 00 00          callq  1b9 <step1+0x1f>
                        1b5: R_X86_64_PLT32     printf-0x4
 1b9:   48 8d 05 00 00 00 00    lea    0x0(%rip),%rax        # 1c0 <step1+0x26>
                        1bc: R_X86_64_PC32      un_initialized-0x4
 1c0:   48 89 c6                mov    %rax,%rsi
 1c3:   48 8d 3d 00 00 00 00    lea    0x0(%rip),%rdi        # 1ca <step1+0x30>
                        1c6: R_X86_64_PC32      .rodata+0xac
 1ca:   b8 00 00 00 00          mov    $0x0,%eax
 1cf:   e8 00 00 00 00          callq  1d4 <step1+0x3a>
                        1d0: R_X86_64_PLT32     printf-0x4
 1d4:   48 8d 05 00 00 00 00    lea    0x0(%rip),%rax        # 1db <step1+0x41>
                        1d7: R_X86_64_PC32      un_init_ptr1-0x4
 1db:   48 89 c6                mov    %rax,%rsi
 1de:   48 8d 3d 00 00 00 00    lea    0x0(%rip),%rdi        # 1e5 <step1+0x4b>
                        1e1: R_X86_64_PC32      .rodata+0xcc
 1e5:   b8 00 00 00 00          mov    $0x0,%eax
 1ea:   e8 00 00 00 00          callq  1ef <step1+0x55>
                        1eb: R_X86_64_PLT32     printf-0x4
 1ef:   48 8d 05 00 00 00 00    lea    0x0(%rip),%rax        # 1f6 <step1+0x5c>
```

```
                        1f2: R_X86_64_PC32un_init_ptr2-0x4
1f6:    48 89 c6                mov    %rax,%rsi
1f9:    48 8d 3d 00 00 00 00    lea    0x0(%rip),%rdi        # 200 <step1+0x66>
                        1fc: R_X86_64_PC32.rodata+0xea
200:    b8 00 00 00 00          mov    $0x0,%eax
205:    e8 00 00 00 00          callq  20a <step1+0x70>
                        206: R_X86_64_PLT32        printf-0x4
20a:    48 8d 05 00 00 00 00    lea    0x0(%rip),%rax        # 211 <step1+0x77>
                        20d: R_X86_64_PC32         init_const-0x4
211:    48 89 c6                mov    %rax,%rsi
214:    48 8d 3d 00 00 00 00    lea    0x0(%rip),%rdi        # 21b <step1+0x81>
                        217: R_X86_64_PC32         .rodata+0x108
21b:    b8 00 00 00 00          mov    $0x0,%eax
220:    e8 00 00 00 00          callq  225 <step1+0x8b>
                        221: R_X86_64_PLT32        printf-0x4
225:    48 8d 05 00 00 00 00    lea    0x0(%rip),%rax        # 22c <step1+0x92>
                        228: R_X86_64_PC32         main-0x4
22c:    48 89 c6                mov    %rax,%rsi
22f:    48 8d 3d 00 00 00 00    lea    0x0(%rip),%rdi        # 236 <step1+0x9c>
                        232: R_X86_64_PC32         .rodata+0x121
236:    b8 00 00 00 00          mov    $0x0,%eax
23b:    e8 00 00 00 00          callq  240 <step1+0xa6>
                        23c: R_X86_64_PLT32        printf-0x4
240:    48 8d 05 00 00 00 00    lea    0x0(%rip),%rax        # 247 <step1+0xad>
                        243: R_X86_64_PC32         step1-0x4
247:    48 89 c6                mov    %rax,%rsi
24a:    48 8d 3d 00 00 00 00    lea    0x0(%rip),%rdi        # 251 <step1+0xb7>
                        24d: R_X86_64_PC32         .rodata+0x133
251:    b8 00 00 00 00          mov    $0x0,%eax
256:    e8 00 00 00 00          callq  25b <step1+0xc1>
                        257: R_X86_64_PLT32        printf-0x4
25b:    48 8d 05 00 00 00 00    lea    0x0(%rip),%rax        # 262 <step1+0xc8>
                        25e: R_X86_64_PC32         step2-0x4
262:    48 89 c6                mov    %rax,%rsi
265:    48 8d 3d 00 00 00 00    lea    0x0(%rip),%rdi        # 26c <step1+0xd2>
                        268: R_X86_64_PC32         .rodata+0x146
26c:    b8 00 00 00 00          mov    $0x0,%eax
271:    e8 00 00 00 00          callq  276 <step1+0xdc>
                        272: R_X86_64_PLT32        printf-0x4
```

```
276:   48 8d 05 00 00 00 00  lea    0x0(%rip),%rax        # 27d <step1+0xe3>
                       279: R_X86_64_PC32          step3-0x4
27d:   48 89 c6              mov    %rax,%rsi
280:   48 8d 3d 00 00 00 00  lea    0x0(%rip),%rdi        # 287 <step1+0xed>
                       283: R_X86_64_PC32          .rodata+0x159
287:   b8 00 00 00 00        mov    $0x0,%eax
28c:   e8 00 00 00 00        callq  291 <step1+0xf7>
                       28d: R_X86_64_PLT32         printf-0x4
291:   48 8d 05 00 00 00 00  lea    0x0(%rip),%rax        # 298 <step1+0xfe>
                       294: R_X86_64_PC32          step4-0x4
298:   48 89 c6              mov    %rax,%rsi
29b:   48 8d 3d 00 00 00 00  lea    0x0(%rip),%rdi        # 2a2 <step1+0x108>
                       29e: R_X86_64_PC32          .rodata+0x16c
2a2:   b8 00 00 00 00        mov    $0x0,%eax
2a7:   e8 00 00 00 00        callq  2ac <step1+0x112>
                       2a8: R_X86_64_PLT32         printf-0x4
2ac:   48 8d 05 00 00 00 00  lea    0x0(%rip),%rax        # 2b3 <step1+0x119>
                       2af: R_X86_64_PC32step5-0x4
2b3:   48 89 c6              mov    %rax,%rsi
2b6:   48 8d 3d 00 00 00 00  lea    0x0(%rip),%rdi        # 2bd <step1+0x123>
                       2b9: R_X86_64_PC32          .rodata+0x17f
2bd:   b8 00 00 00 00        mov    $0x0,%eax
2c2:   e8 00 00 00 00        callq  2c7 <step1+0x12d>
                       2c3: R_X86_64_PLT32         printf-0x4
2c7:   48 8d 05 00 00 00 00  lea    0x0(%rip),%rax        # 2ce <step1+0x134>
                       2ca: R_X86_64_PC32          step6-0x4
2ce:   48 89 c6              mov    %rax,%rsi
2d1:   48 8d 3d 00 00 00 00  lea    0x0(%rip),%rdi        # 2d8 <step1+0x13e>
                       2d4: R_X86_64_PC32          .rodata+0x192
2d8:   b8 00 00 00 00        mov    $0x0,%eax
2dd:   e8 00 00 00 00        callq  2e2 <step1+0x148>
                       2de: R_X86_64_PLT32         printf-0x4
2e2:   48 8b 05 00 00 00 00  mov    0x0(%rip),%rax        # 2e9 <step1+0x14f>
                       2e5: R_X86_64_REX_GOTPCRELX          foo-0x4
2e9:   48 89 c6              mov    %rax,%rsi
2ec:   48 8d 3d 00 00 00 00  lea    0x0(%rip),%rdi        # 2f3 <step1+0x159>
                       2ef: R_X86_64_PC32.rodata+0x1a5
2f3:   b8 00 00 00 00        mov    $0x0,%eax
2f8:   e8 00 00 00 00        callq  2fd <step1+0x163>
```

```
                        2f9: R_X86_64_PLT32         printf-0x4
 2fd:   48 8b 05 00 00 00 00  mov    0x0(%rip),%rax        # 304 <step1+0x16a>
                        300: R_X86_64_REX_GOTPCRELX         printf-0x4
 304:   48 89 c6              mov    %rax,%rsi
 307:   48 8d 3d 00 00 00 00  lea    0x0(%rip),%rdi        # 30e <step1+0x174>
                        30a: R_X86_64_PC32          .rodata+0x1b6
 30e:   b8 00 00 00 00        mov    $0x0,%eax
 313:   e8 00 00 00 00        callq  318 <step1+0x17e>
                        314: R_X86_64_PLT32         printf-0x4
 318:   90                    nop
 319:   5d                    pop    %rbp
 31a:   c3                    retq

000000000000031b <step2>:
 31b:   55                    push   %rbp
 31c:   48 89 e5              mov    %rsp,%rbp
 31f:   bf 80 84 1e 00        mov    $0x1e8480,%edi
 324:   e8 00 00 00 00        callq  329 <step2+0xe>
                        325: R_X86_64_PLT32         malloc-0x4
 329:   48 89 05 00 00 00 00  mov    %rax,0x0(%rip)        # 330 <step2+0x15>
                        32c: R_X86_64_PC32          un_init_ptr1-0x4
 330:   bf 80 84 1e 00        mov    $0x1e8480,%edi
 335:   e8 00 00 00 00        callq  33a <step2+0x1f>
                        336: R_X86_64_PLT32         malloc-0x4
 33a:   48 89 05 00 00 00 00  mov    %rax,0x0(%rip)        # 341 <step2+0x26>
                        33d: R_X86_64_PC32          un_init_ptr2-0x4
 341:   48 8b 05 00 00 00 00  mov    0x0(%rip),%rax        # 348 <step2+0x2d>
                        344: R_X86_64_PC32          un_init_ptr1-0x4
 348:   48 89 c6              mov    %rax,%rsi
 34b:   48 8d 3d 00 00 00 00  lea    0x0(%rip),%rdi        # 352 <step2+0x37>
                        34e: R_X86_64_PC32          .rodata+0x1cc
 352:   b8 00 00 00 00        mov    $0x0,%eax
 357:   e8 00 00 00 00        callq  35c <step2+0x41>
                        358: R_X86_64_PLT32         printf-0x4
 35c:   48 8b 05 00 00 00 00  mov    0x0(%rip),%rax        # 363 <step2+0x48>
                        35f: R_X86_64_PC32un_init_ptr2-0x4
 363:   48 89 c6              mov    %rax,%rsi
 366:   48 8d 3d 00 00 00 00  lea    0x0(%rip),%rdi        # 36d <step2+0x52>
                        369: R_X86_64_PC32          .rodata+0x1f4
 36d:   b8 00 00 00 00          mov    $0x0,%eax
 372:   e8 00 00 00 00          callq  377 <step2+0x5c>
                        373: R_X86_64_PLT32     printf-0x4
 377:   90                      nop
 378:   5d                      pop    %rbp
 379:   c3                      retq

000000000000037a <step3>:
 37a:   55                      push   %rbp
 37b:   48 89 e5                mov    %rsp,%rbp
 37e:   bf d4 30 00 00          mov    $0x30d4,%edi
 383:   b8 00 00 00 00          mov    $0x0,%eax
 388:   e8 00 00 00 00          callq  38d <step3+0x13>
                        389: R_X86_64_PLT32     foo-0x4
 38d:   90                      nop
 38e:   5d                      pop    %rbp
 38f:   c3                      retq

0000000000000390 <step4>:
 390:   55                      push   %rbp
 391:   48 89 e5                mov    %rsp,%rbp
 394:   48 83 ec 10             sub    $0x10,%rsp
 398:   48 8d 35 00 00 00 00    lea    0x0(%rip),%rsi        # 39f <step4+0xf>
                        39b: R_X86_64_PC32      .rodata+0x216
 39f:   48 8d 3d 00 00 00 00    lea    0x0(%rip),%rdi        # 3a6 <step4+0x16>
                        3a2: R_X86_64_PC32      .rodata+0x218
 3a6:   e8 00 00 00 00          callq  3ab <step4+0x1b>
                        3a7: R_X86_64_PLT32     fopen-0x4
 3ab:   48 89 45 f8             mov    %rax,-0x8(%rbp)
 3af:   48 8b 45 f8             mov    -0x8(%rbp),%rax
 3b3:   ba 00 00 00 00          mov    $0x0,%edx
 3b8:   be 80 84 1e 00          mov    $0x1e8480,%esi
 3bd:   48 89 c7                mov    %rax,%rdi
 3c0:   e8 00 00 00 00          callq  3c5 <step4+0x35>
                        3c1: R_X86_64_PLT32     fseek-0x4
 3c5:   48 8b 45 f8             mov    -0x8(%rbp),%rax
 3c9:   48 89 c6                mov    %rax,%rsi
 3cc:   bf 00 00 00 00          mov    $0x0,%edi
 3d1:   e8 00 00 00 00          callq  3d6 <step4+0x46>
```

```
                        3d2: R_X86_64_PLT32     fputc-0x4
 3d6:   48 8b 45 f8             mov    -0x8(%rbp),%rax
 3da:   48 89 c7                mov    %rax,%rdi
 3dd:   e8 00 00 00 00          callq  3e2 <step4+0x52>
                        3de: R_X86_64_PLT32     fclose-0x4
 3e2:   be 02 00 00 00          mov    $0x2,%esi
 3e7:   48 8d 3d 00 00 00 00    lea    0x0(%rip),%rdi        # 3ee <step4+0x5e>
                        3ea: R_X86_64_PC32      .rodata+0x218
 3ee:   b8 00 00 00 00          mov    $0x0,%eax
 3f3:   e8 00 00 00 00          callq  3f8 <step4+0x68>
                        3f4: R_X86_64_PLT32     open-0x4
 3f8:   89 45 f4                mov    %eax,-0xc(%rbp)
 3fb:   8b 45 f4                mov    -0xc(%rbp),%eax
 3fe:   41 b9 00 00 00 00       mov    $0x0,%r9d
 404:   41 89 c0                mov    %eax,%r8d
 407:   b9 02 00 00 00          mov    $0x2,%ecx
 40c:   ba 03 00 00 00          mov    $0x3,%edx
 411:   be 80 84 1e 00          mov    $0x1e8480,%esi
 416:   bf 00 00 00 00          mov    $0x0,%edi
 41b:   e8 00 00 00 00          callq  420 <step4+0x90>
                        41c: R_X86_64_PLT32     mmap-0x4
 420:   48 89 05 00 00 00 00    mov    %rax,0x0(%rip)        # 427 <step4+0x97>
                        423: R_X86_64_PC32      region-0x4
 427:   48 8b 05 00 00 00 00    mov    0x0(%rip),%rax        # 42e <step4+0x9e>
                        42a: R_X86_64_PC32      region-0x4
 42e:   48 89 c6                mov    %rax,%rsi
 431:   48 8d 3d 00 00 00 00    lea    0x0(%rip),%rdi        # 438 <step4+0xa8>
                        434: R_X86_64_PC32      .rodata+0x220
 438:   b8 00 00 00 00          mov    $0x0,%eax
 43d:   e8 00 00 00 00          callq  442 <step4+0xb2>
                        43e: R_X86_64_PLT32     printf-0x4
 442:   90                      nop
 443:   c9                      leaveq
 444:   c3                      retq

0000000000000445 <step5>:
 445:   55                      push   %rbp
 446:   48 89 e5                mov    %rsp,%rbp
 449:   c7 45 fc 00 00 00 00    movl   $0x0,-0x4(%rbp)
 450:   eb 16                   jmp    468 <step5+0x23>
 452:   48 8b 15 00 00 00 00    mov    0x0(%rip),%rdx        # 459 <step5+0x14>
                        455: R_X86_64_PC32      region-0x4
 459:   8b 45 fc                mov    -0x4(%rbp),%eax
 45c:   48 98                   cltq
 45e:   48 01 d0                add    %rdx,%rax
 461:   c6 00 6c                movb   $0x6c,(%rax)
 464:   83 45 fc 01             addl   $0x1,-0x4(%rbp)
 468:   81 7d fc 6f 17 00 00    cmpl   $0x176f,-0x4(%rbp)
 46f:   7e e1                   jle    452 <step5+0xd>
 471:   90                      nop
 472:   5d                      pop    %rbp
 473:   c3                      retq

0000000000000474 <step6>:
 474:   55                      push   %rbp
 475:   48 89 e5                mov    %rsp,%rbp
 478:   48 83 ec 10             sub    $0x10,%rsp
 47c:   48 c7 45 f8 00 00 40    movq   $0x400000,-0x8(%rbp)
 483:   00
 484:   c7 45 f4 00 00 00 00    movl   $0x0,-0xc(%rbp)
 48b:   eb 4d                   jmp    4da <step6+0x66>
 48d:   8b 45 f4                mov    -0xc(%rbp),%eax
 490:   48 98                   cltq
 492:   48 8d 14 85 00 00 00    lea    0x0(,%rax,4),%rdx
 499:   00
 49a:   48 8b 45 f8             mov    -0x8(%rbp),%rax
 49e:   48 01 d0                add    %rdx,%rax
 4a1:   8b 00                   mov    (%rax),%eax
 4a3:   89 c7                   mov    %eax,%edi
 4a5:   e8 00 00 00 00          callq  4aa <step6+0x36>
                        4a6: R_X86_64_PLT32     htonl-0x4
 4aa:   89 c1                   mov    %eax,%ecx
 4ac:   8b 45 f4                mov    -0xc(%rbp),%eax
 4af:   48 98                   cltq
 4b1:   48 8d 14 85 00 00 00    lea    0x0(,%rax,4),%rdx
 4b8:   00
 4b9:   48 8b 45 f8             mov    -0x8(%rbp),%rax
 4bd:   48 01 d0                add    %rdx,%rax
 4c0:   89 ca                   mov    %ecx,%edx
 4c2:   48 89 c6                mov    %rax,%rsi
 4c5:   48 8d 3d 00 00 00 00    lea    0x0(%rip),%rdi        # 4cc <step6+0x58>
                        4c8: R_X86_64_PC32          .rodata+0x23d
 4cc:   b8 00 00 00 00          mov    $0x0,%eax
 4d1:   e8 00 00 00 00          callq  4d6 <step6+0x62>
                        4d2: R_X86_64_PLT32         printf-0x4
 4d6:   83 45 f4 01             addl   $0x1,-0xc(%rbp)
 4da:   81 7d f4 ff 03 00 00    cmpl   $0x3ff,-0xc(%rbp)
 4e1:   7e aa                   jle    48d <step6+0x19>
 4e3:   90                      nop
 4e4:   c9                      leaveq
 4e5:   c3                      retq
```

OBJDUMP of module2.o:

```
module2.o:     file format elf64-x86-64
module2.o
architecture: i386:x86-64, flags 0x00000010:
HAS_SYMS
start address 0x0000000000000000

Sections:
Idx Name          Size      VMA               LMA               File off  Algn
  0 .text         00000000  0000000000000000  0000000000000000  00000040  2**0
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000080  0000000000000000  0000000000000000  00000040  2**5
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  0000000000000000  0000000000000000  000000c0  2**0
                  ALLOC
  3 .rodata       00000004  0000000000000000  0000000000000000  000000c0  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .comment      0000002a  0000000000000000  0000000000000000  000000c4  2**0
                  CONTENTS, READONLY
  5 .note.GNU-stack 00000000  0000000000000000  0000000000000000  000000ee  2**0
                  CONTENTS, READONLY
SYMBOL TABLE:
0000000000000000 l    df *ABS*      0000000000000000 module2.c
0000000000000000 l    d  .text      0000000000000000 .text
0000000000000000 l    d  .data      0000000000000000 .data
```

```
0000000000000000 l    d  .bss           0000000000000000 .bss
0000000000000000 l    d  .rodata        0000000000000000 .rodata
0000000000000000 l    d  .note.GNU-stack   0000000000000000 .note.GNU-stack
0000000000000000 l    d  .comment  0000000000000000 .comment
0000000000000000 g     O .data          0000000000000080 initialized
0000000000003e80       O *COM*    0000000000000020 un_initialized
0000000000000008       O *COM*    0000000000000008 un_init_ptr1
0000000000000008       O *COM*    0000000000000008 un_init_ptr2
0000000000000000 g     O .rodata      0000000000000004 init_const
```

OBJDUMP of module3.o:

```
module3.o:     file format elf64-x86-64
module3.o
architecture: i386:x86-64, flags 0x00000011:
HAS_RELOC, HAS_SYMS
start address 0x0000000000000000

Sections:
Idx Name          Size      VMA               LMA               File off  Algn
  0 .text         00000055  0000000000000000  0000000000000000  00000040  2**0
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  0000000000000000  0000000000000000  00000095  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  0000000000000000  0000000000000000  00000095  2**0
                  ALLOC
  3 .rodata       00000023  0000000000000000  0000000000000000  00000098  2**3
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .comment      0000002a  0000000000000000  0000000000000000  000000bb  2**0
                  CONTENTS, READONLY
  5 .note.GNU-stack 00000000  0000000000000000  0000000000000000  000000e5  2**0
                  CONTENTS, READONLY
  6 .eh_frame     00000038  0000000000000000  0000000000000000  000000e8  2**3
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, DATA
SYMBOL TABLE:
0000000000000000 l    df *ABS*      0000000000000000 module3.c
0000000000000000 l    d  .text          0000000000000000 .text
0000000000000000 l    d  .data          0000000000000000 .data
0000000000000000 l    d  .bss           0000000000000000 .bss
0000000000000000 l    d  .rodata        0000000000000000 .rodata
```

```
0000000000000000 l    d  .note.GNU-stack   0000000000000000 .note.GNU-stack
0000000000000000 l    d  .eh_frame  0000000000000000 .eh_frame
0000000000000000 l    d  .comment  0000000000000000 .comment
0000000000000000 g     F .text         0000000000000055 foo
0000000000000000         *UND*      0000000000000000 _GLOBAL_OFFSET_TABLE_
0000000000000000         *UND*      0000000000000000 printf



Disassembly of section .text:

0000000000000000 <foo>:
   0:     55                    push   %rbp
   1:     48 89 e5                  mov    %rsp,%rbp
   4:     48 83 ec 20               sub    $0x20,%rsp
   8:     89 7d ec                  mov    %edi,-0x14(%rbp)
   b:     8b 45 ec                  mov    -0x14(%rbp),%eax
   e:     85 c0                     test   %eax,%eax
  10:     7e 40                     jle    52 <foo+0x52>
  12:     48 c7 45 f8 01 00 00  movq   $0x1,-0x8(%rbp)
  19:     00
  1a:     8b 45 ec                  mov    -0x14(%rbp),%eax
  1d:     3d d4 30 00 00            cmp    $0x30d4,%eax
  22:     74 08                     je     2c <foo+0x2c>
  24:     8b 45 ec                  mov    -0x14(%rbp),%eax
  27:     83 f8 01                  cmp    $0x1,%eax
  2a:     75 18                     jne    44 <foo+0x44>
  2c:     48 8d 45 ec               lea    -0x14(%rbp),%rax
  30:     48 89 c6                  mov    %rax,%rsi
  33:     48 8d 3d 00 00 00 00  lea    0x0(%rip),%rdi        # 3a <foo+0x3a>
                        36: R_X86_64_PC32  .rodata-0x4
  3a:     b8 00 00 00 00            mov    $0x0,%eax
  3f:     e8 00 00 00 00            callq  44 <foo+0x44>
                        40: R_X86_64_PLT32           printf-0x4
  44:     8b 45 ec                  mov    -0x14(%rbp),%eax
  47:     83 e8 01                  sub    $0x1,%eax
  4a:     89 c7                     mov    %eax,%edi
  4c:     e8 00 00 00 00            callq  51 <foo+0x51>
                        4d: R_X86_64_PC32  foo-0x4
```

```
51:   90          nop
52:   90          nop
53:   c9          leaveq
54:   c3          retq
```