DEMYSTIFYING PROCESS ADDRESS SPACE

HEAP, STACK, AND BEYOND

Piotr Wierciński
The Qt Company

AGENDA

- introduction
- some theory
- analyzing process on Linux

ABOUT ME

My name is Piotr Wierciński.
I work at the Qt Company in Oslo, at the Web Assembly platform team.
In my free time I'm just trying to figure out how computers and life works.

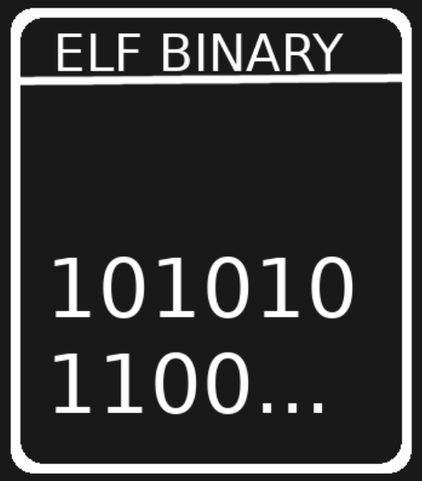
SOCIALS & SLIDES

& @laminowany

Slides: https://github.com/laminowany/Slides

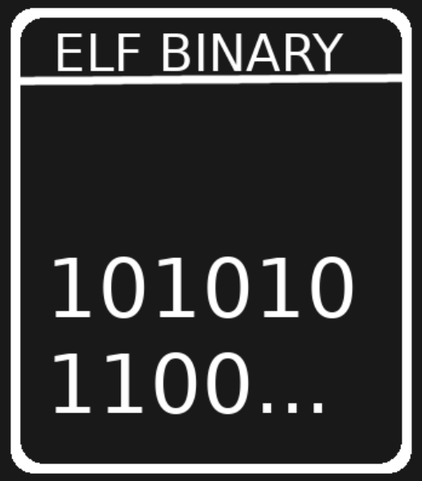
WHAT IS THIS TALK ABOUT?

WHAT IS THIS TALK ABOUT?



APPLICATION

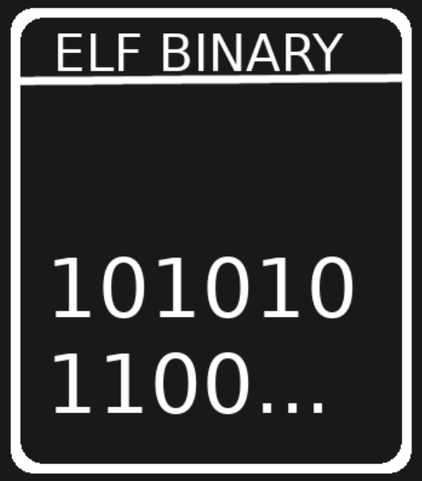
WHAT IS THIS TALK ABOUT?



APPLICATION

OS

WHAT IS THIS TALK ABOUT?

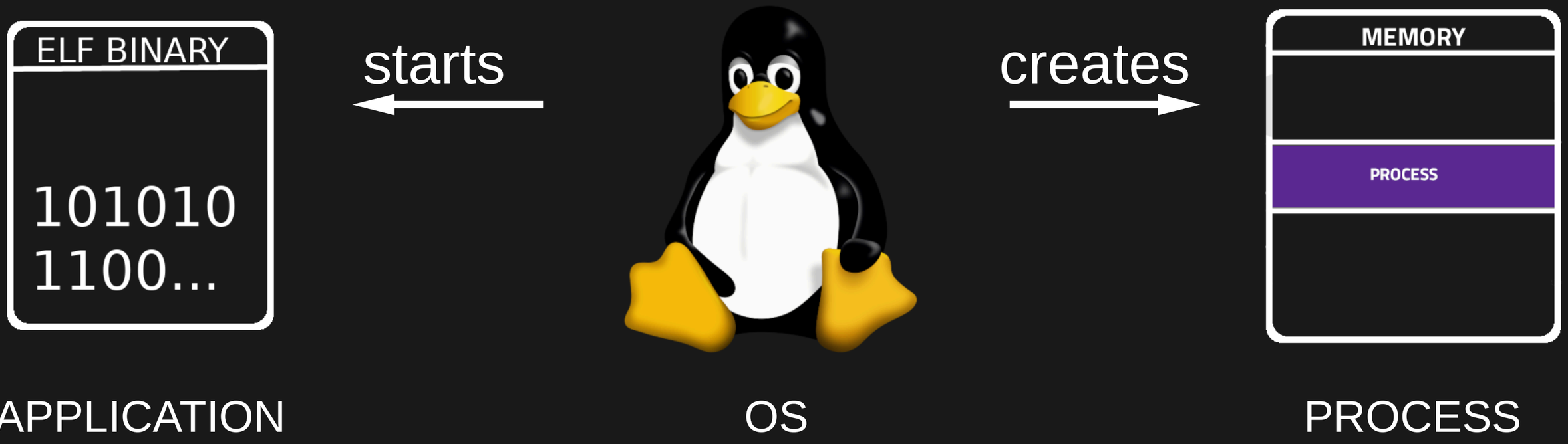


starts

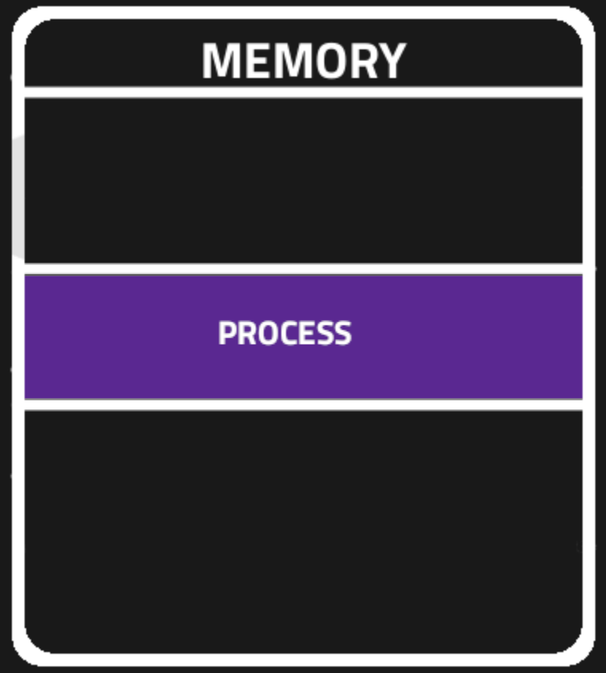
APPLICATION

OS

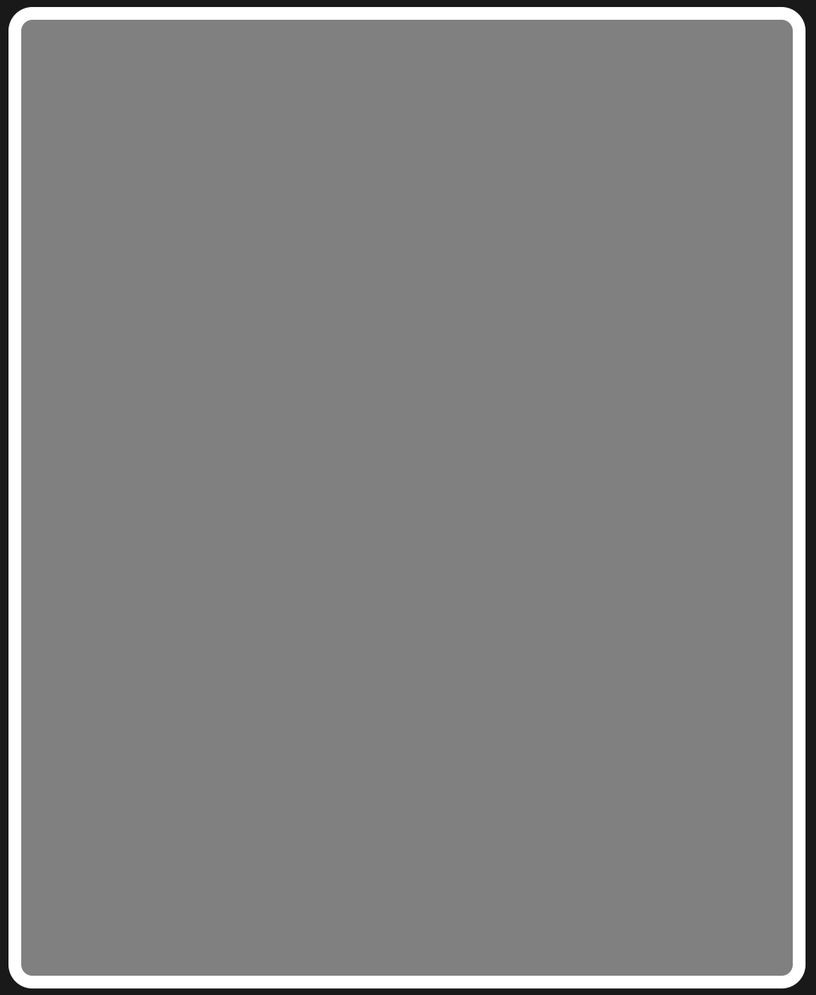
WHAT IS THIS TALK ABOUT?



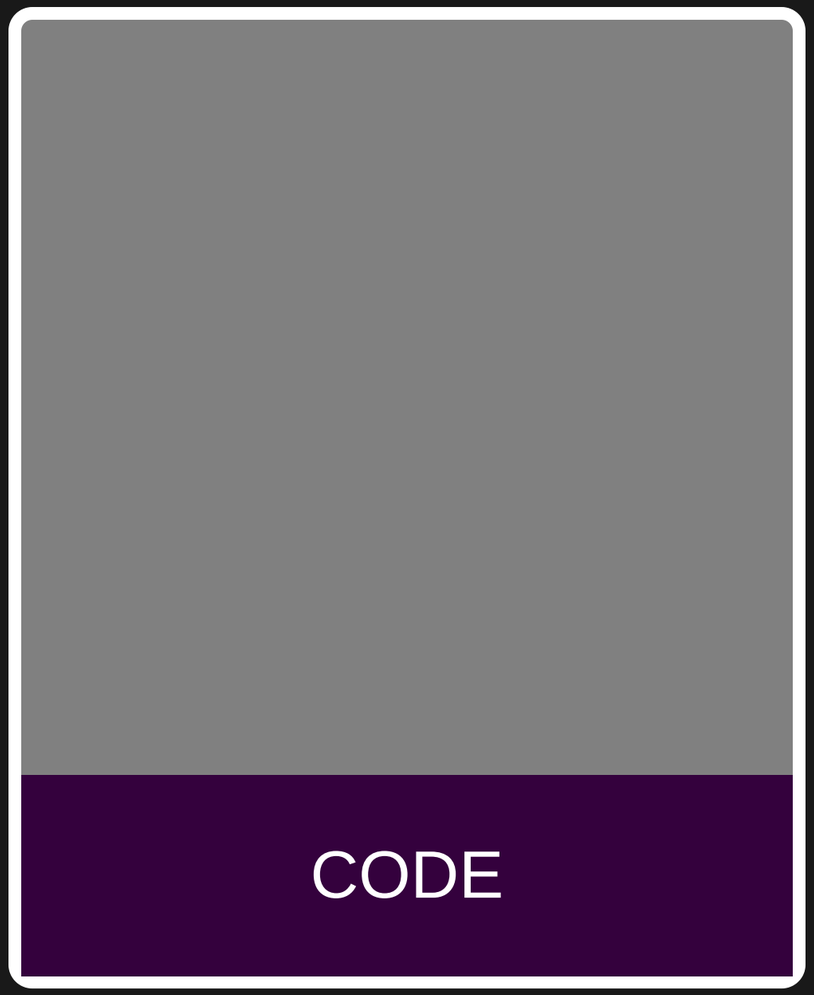
WHAT IS THIS TALK ABOUT?



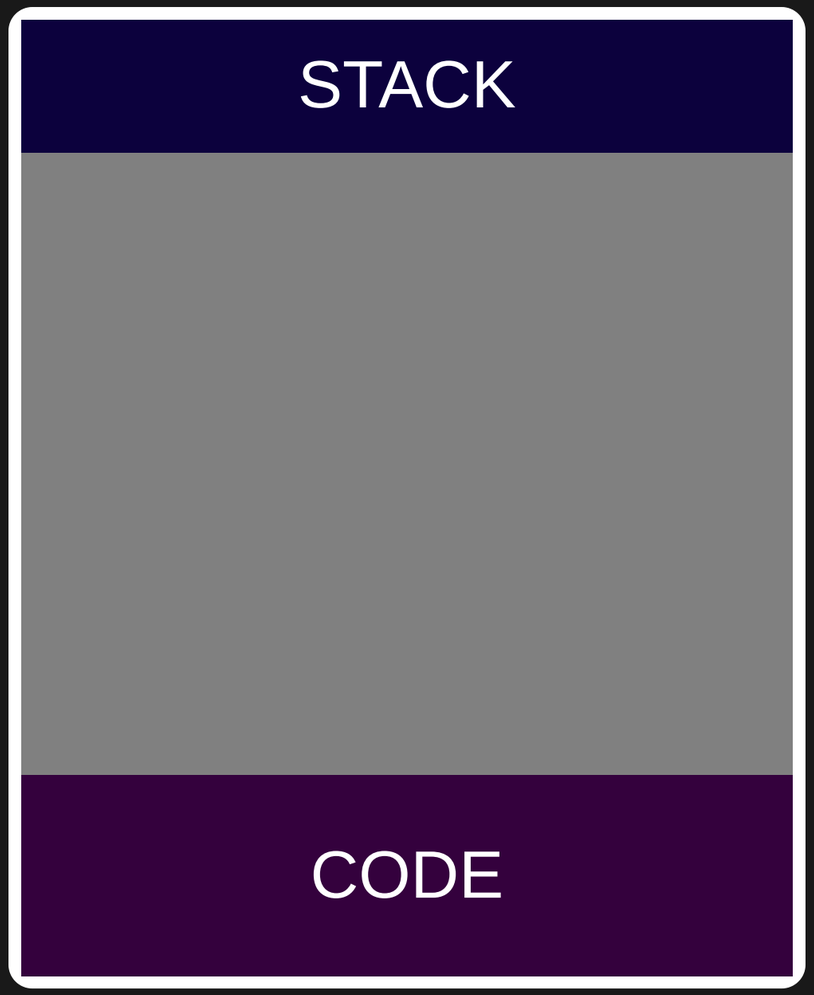
WHAT WAS MY INITIAL MENTAL MODEL?



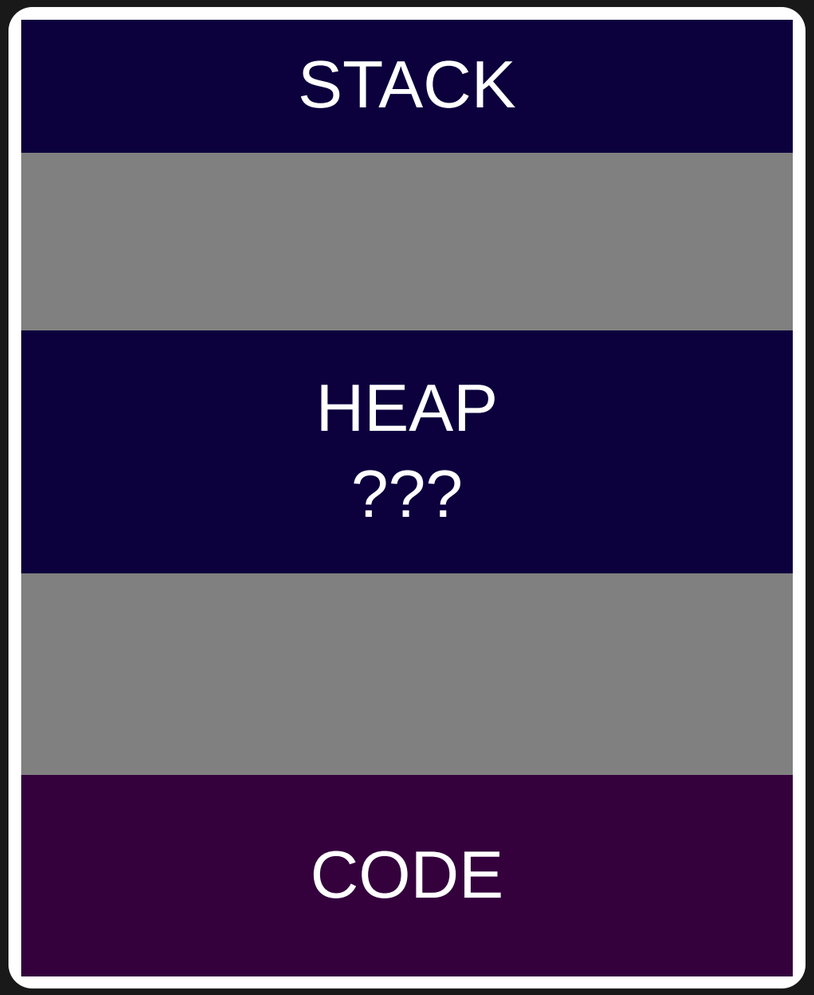
WHAT WAS MY INITIAL MENTAL MODEL?



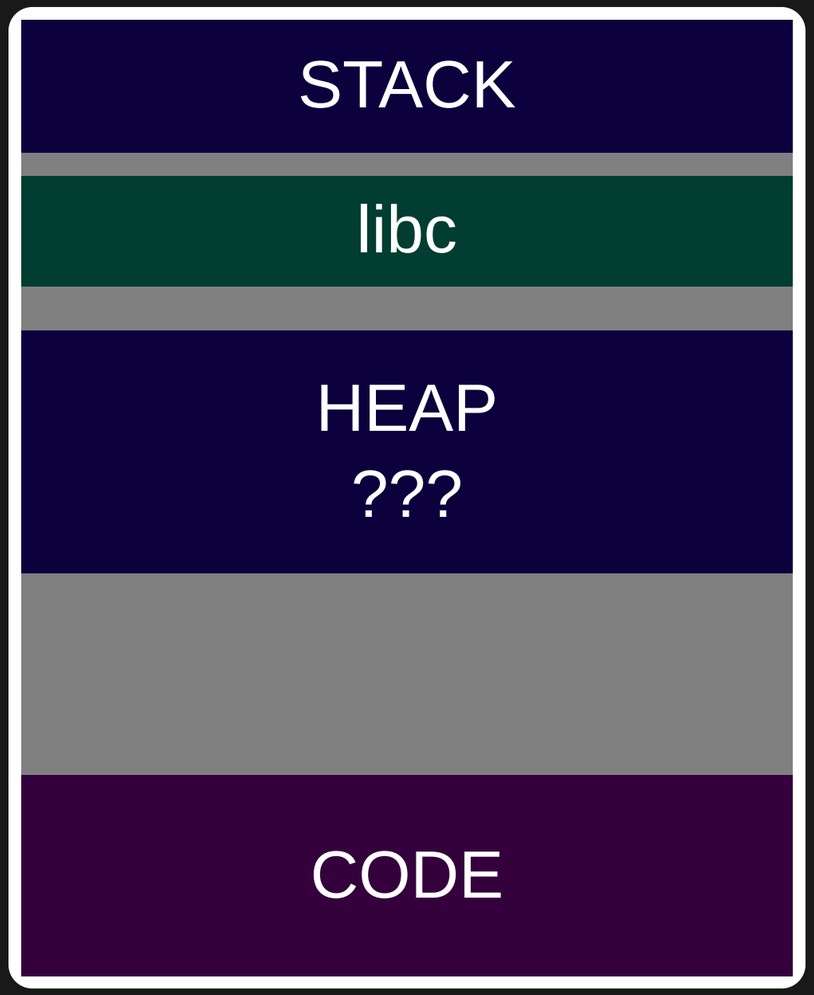
WHAT WAS MY INITIAL MENTAL MODEL?



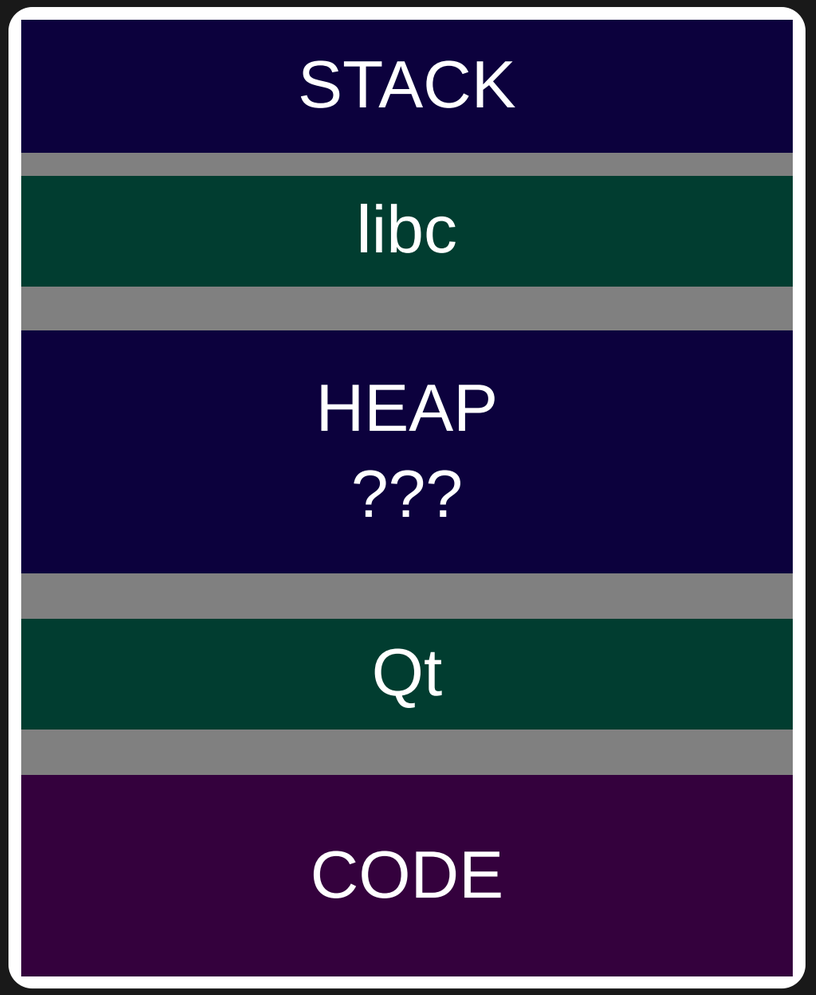
WHAT WAS MY INITIAL MENTAL MODEL?



WHAT WAS MY INITIAL MENTAL MODEL?



WHAT WAS MY INITIAL MENTAL MODEL?

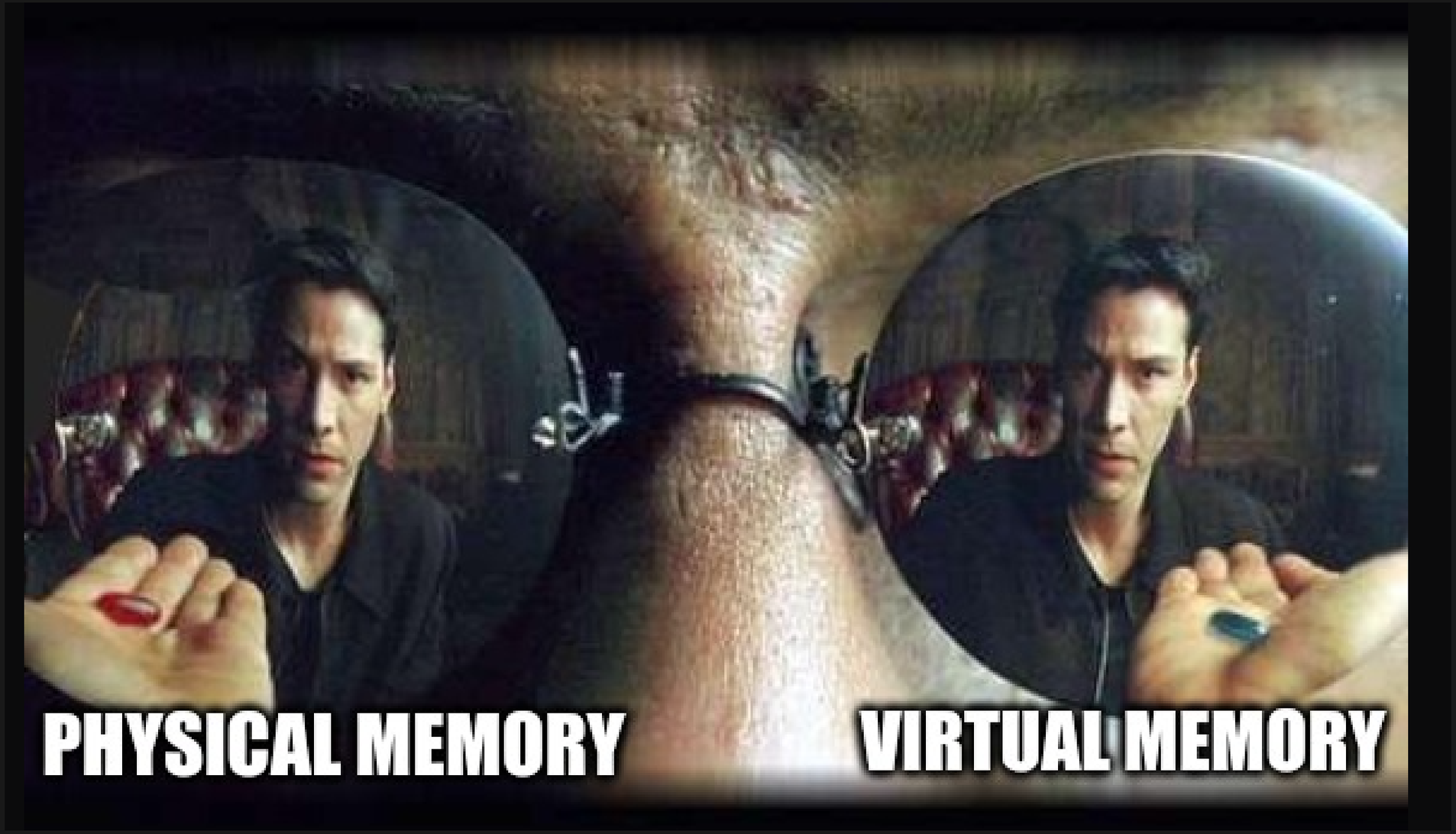


Do I need to know all this as a software engineer?

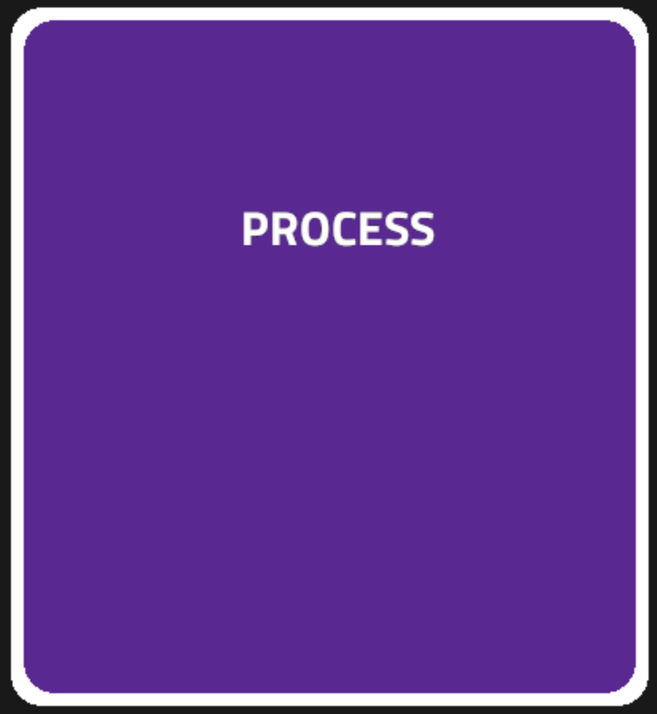
Do I need to know all this as a software engineer?

NO

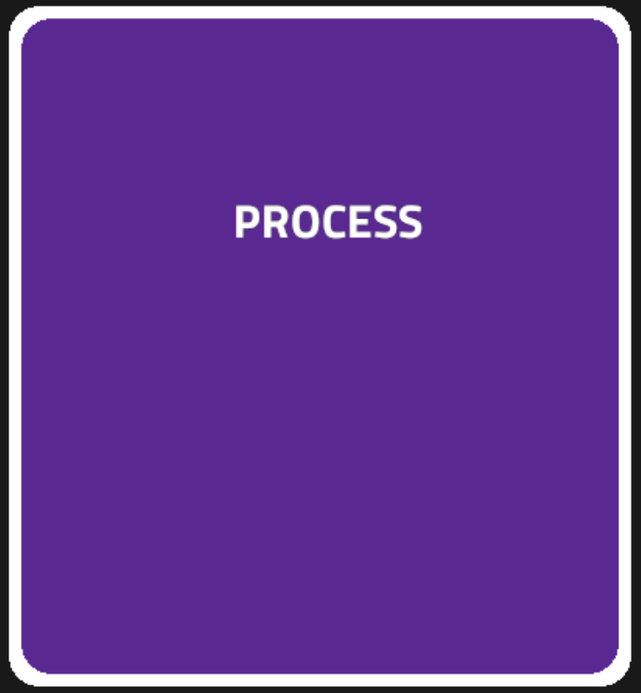
THERE ARE TWO WAYS TO THINK ABOUT MEMORY



VIRTUAL MEMORY ILLUSIONS



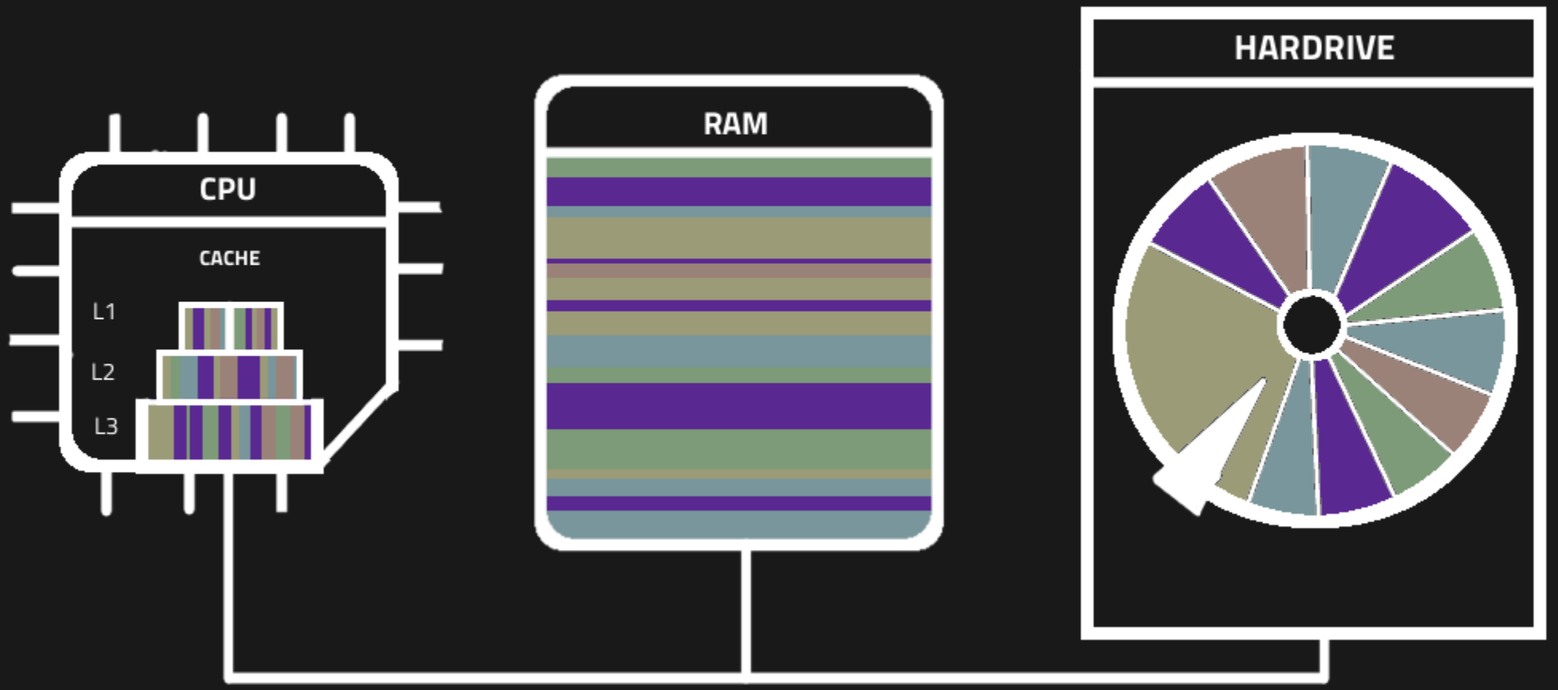
VIRTUAL MEMORY ILLUSIONS



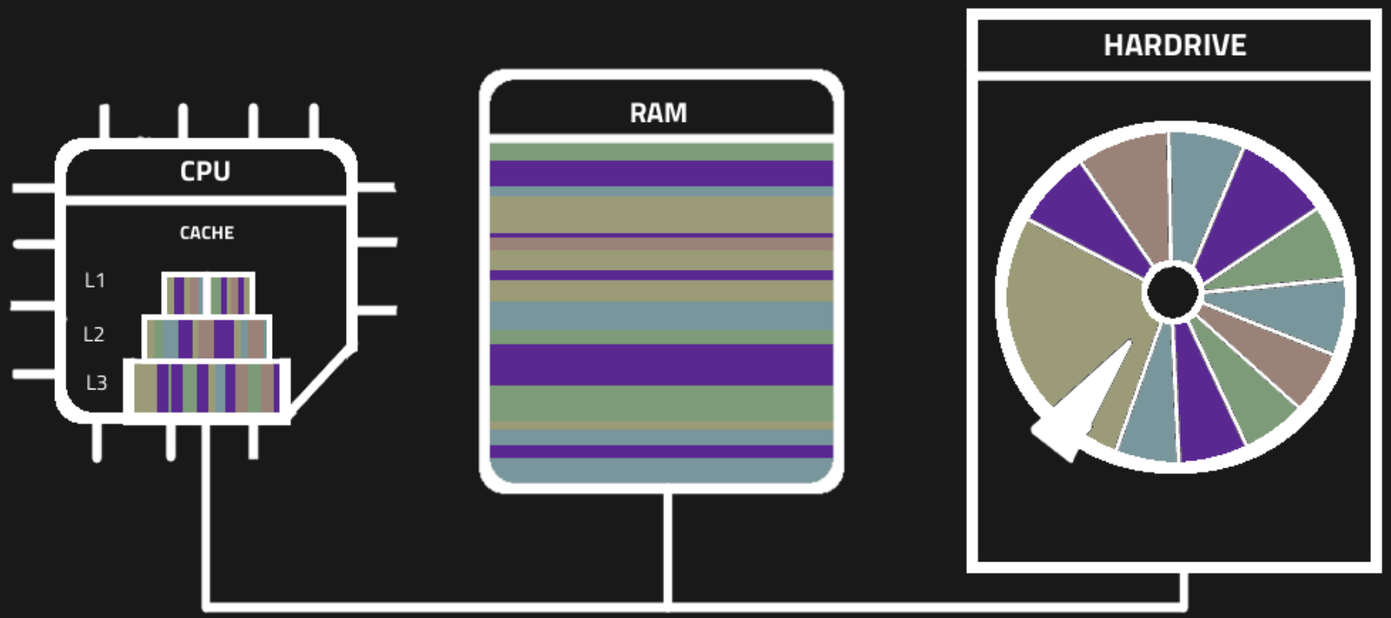
- continuous block of memory, no gaps
- entire address space available to the process
- can be larger than physical memory
- isolated from other processes

PHYSICAL MEMORY REALITY

PHYSICAL MEMORY REALITY

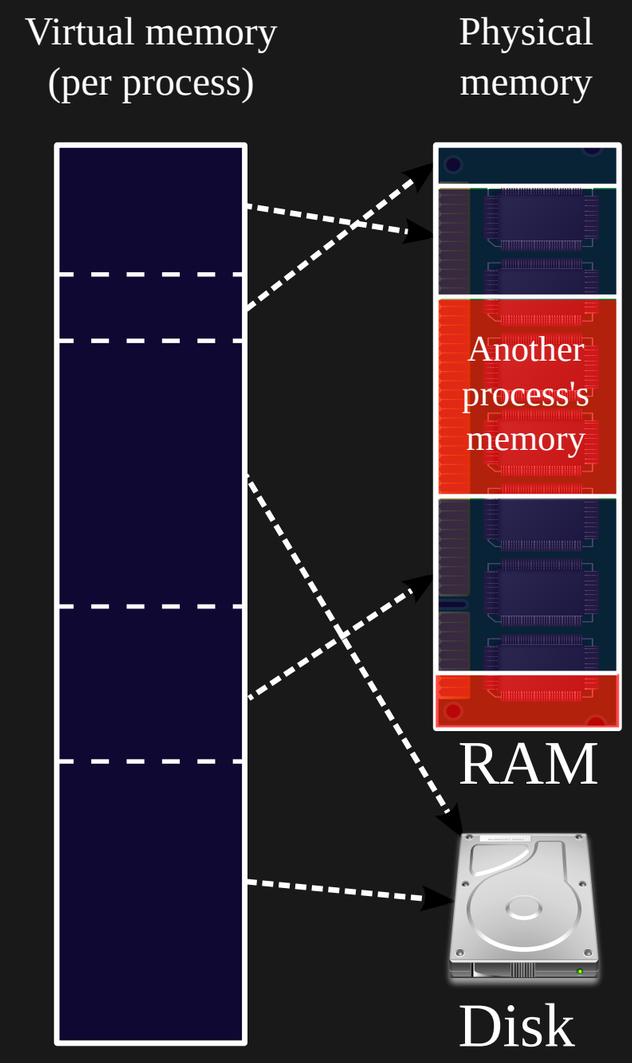


PHYSICAL MEMORY REALITY



- hierarchical
- fragmented
- no isolation between processes
- specialized (L1 cache)

VIRTUAL TO PHYSICAL



Disclaimers:

- x86 architecture
- virtual memory perspective
- focus on Linux
- gdb with `pwndbg` plugin (https://github.com/pwndbg/pwndbg)

HOW BIG IS THE PROCESS ADDRESS SPACE?

HOW BIG IS THE PROCESS ADDRESS SPACE?

32-BIT

2^32 bytes = 4GB

HOW BIG IS THE PROCESS ADDRESS SPACE?

32-BIT

2^32 bytes = 4GB

64-BIT

2^48 bytes = 256TB (256,000GB)

OR

2^57 bytes = 4PB (4,000,000GB)

Why not 2^64 on 64-bit?

- 2^64 (16 exabytes) is still a theoretical limit of architecture
- 48 and 57 bits is what current implementations are supporting (for practical reasons)

Why 48 bits addressing?

- introduced by AMD Opteron in 2003 (first x86-64)
- 256TB was enough back then, and still is
- sensible default

Why 57 bits addressing?

- intended for specialized server CPUs
- often referred to as "Intel 5-level paging"
- supported in Linux since 4.14 (2017) and Windows server editions

Key takeaways:

Key takeaways:

- 32 bits virtual addressing for 32 bits CPU
- 48 bits virtual addressing for 64 bits CPU

Further reading

https://ocw.cs.pub.ro/courses/_media/asc/lab4/opteronoverview.pdf

https://en.wikipedia.org/wiki/Intel_5-level_paging

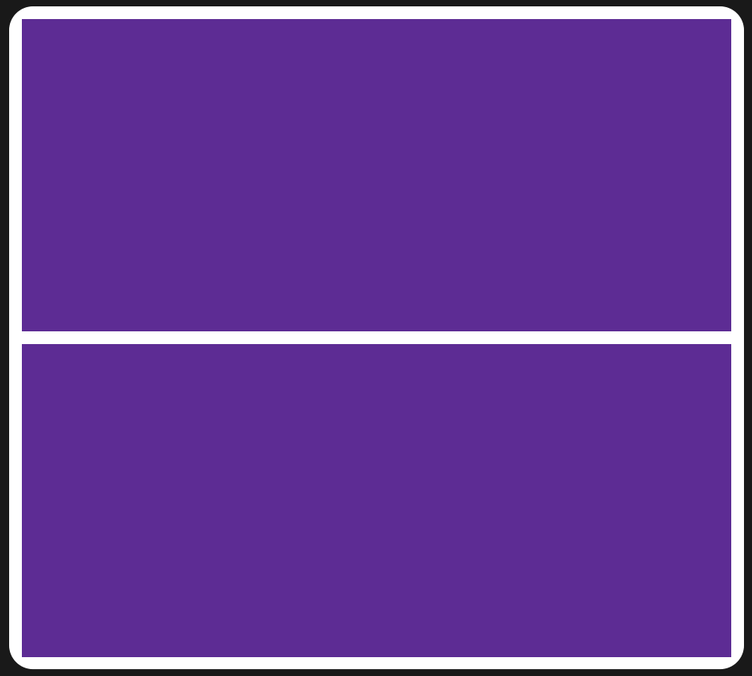
https://docs.kernel.org/next/x86/x86_64/5level-paging.html

https://en.wikipedia.org/wiki/X86-64#Virtual_address_space_details

PROCESS ADDRESS SPACE

PROCESS ADDRESS SPACE

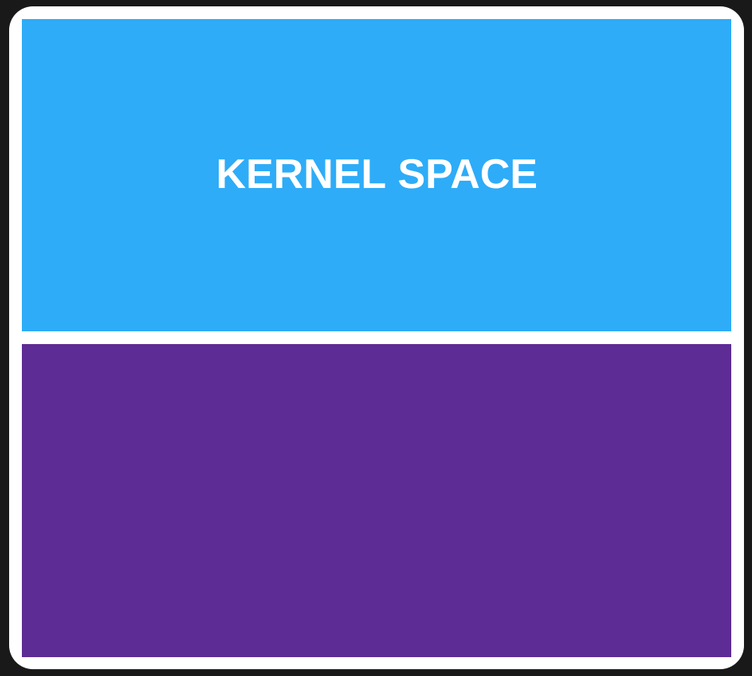
Divided into two parts:



PROCESS ADDRESS SPACE

Divided into two parts:

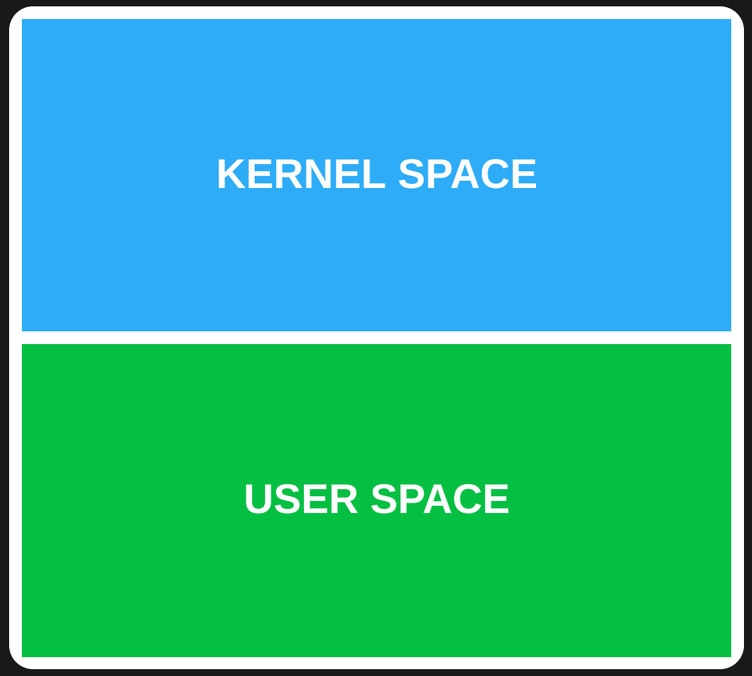
- kernel space



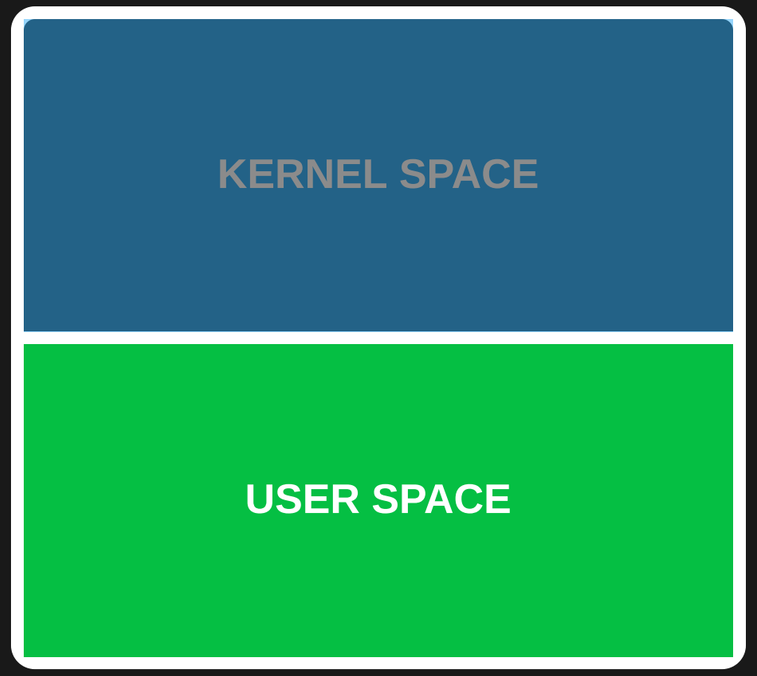
PROCESS ADDRESS SPACE

Divided into two parts:

- kernel space
- user space

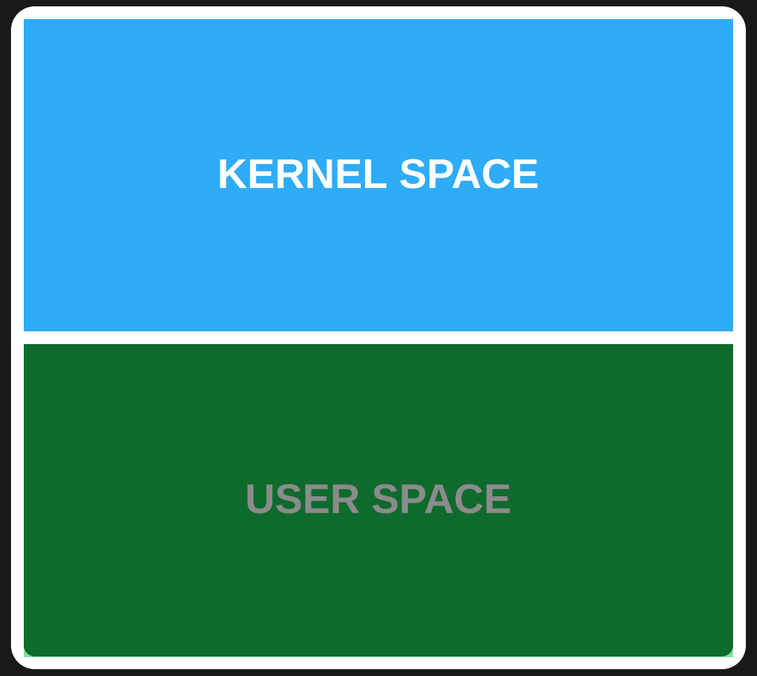


PROCESS ADDRESS SPACE



Contains the program that is being run

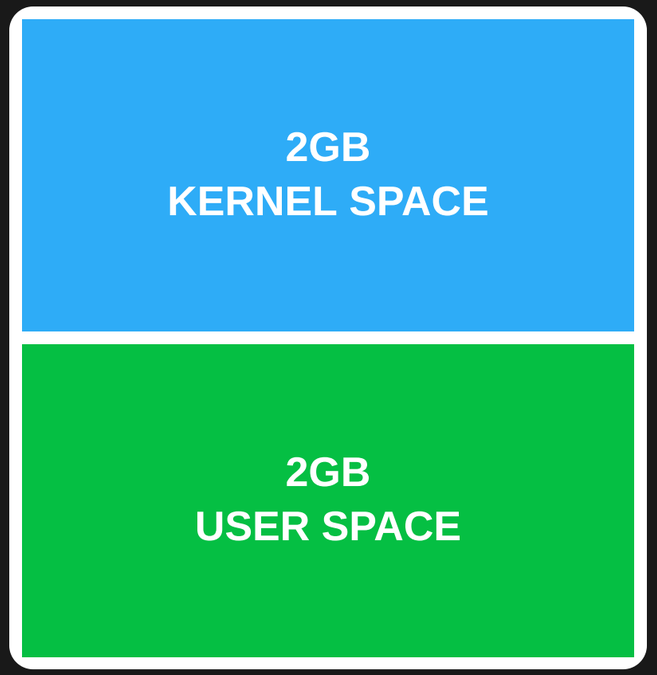
PROCESS ADDRESS SPACE



Contains OS kernel

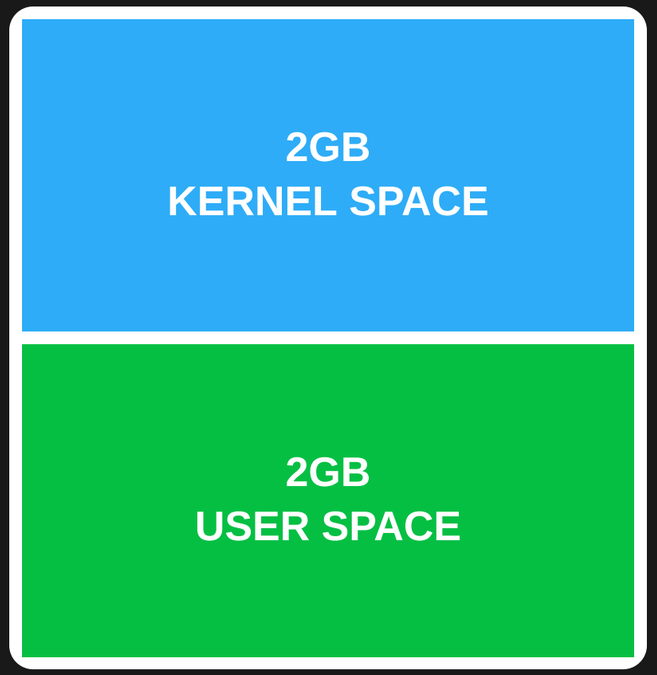
32 BIT SYSTEMS (4GB)

32 BIT SYSTEMS (4GB)

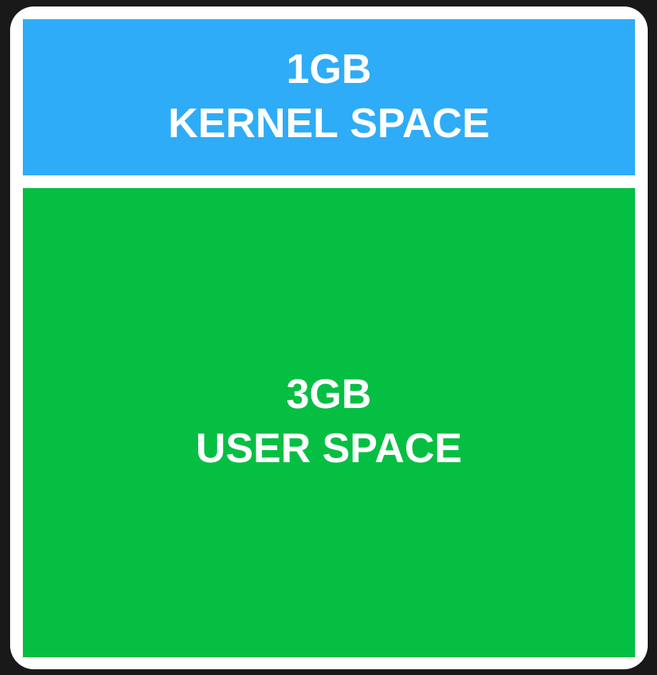


WINDOWS

32 BIT SYSTEMS (4GB)



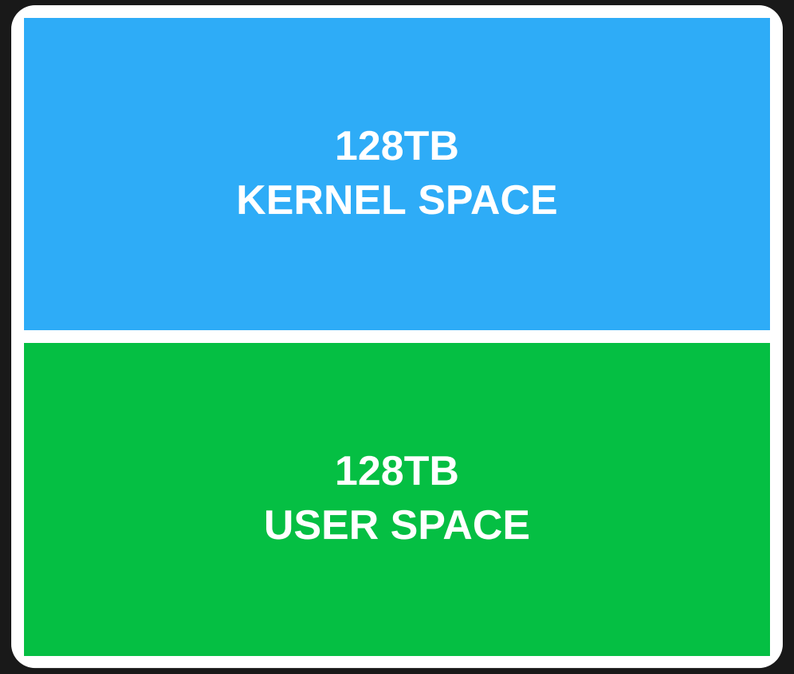
WINDOWS



LINUX
(often called "3/1 split")

64 BIT SYSTEMS (256TB)

64 BIT SYSTEMS (256TB)



LINUX & WINDOWS

Why one address space?

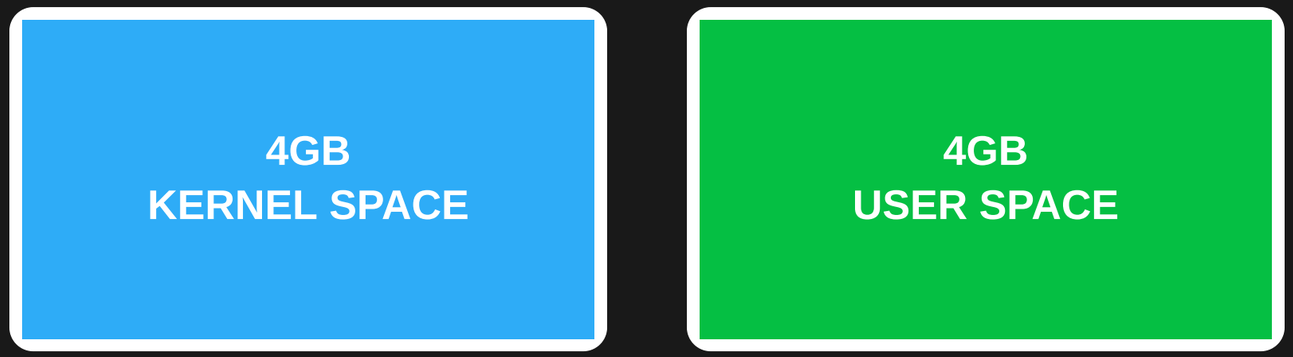
Why one address space?

Performance reasons:

- cheaper mode switch
- cheaper context switch (kernel stays mapped)
- less trashing TLB (Translation Lookaside Buffer)

USER <-> KERNEL SEPARATION

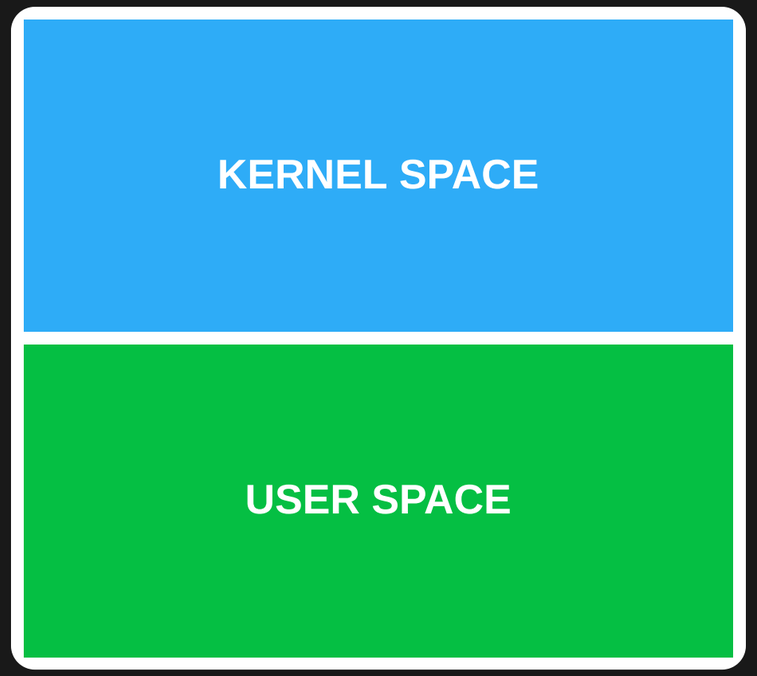
In 2003 there was a "4G/4G split" patch for linux kernel, to seperate kernel and user address space (on 32 bits).



more details: https://lwn.net/Articles/39283/

MODERN LINUX

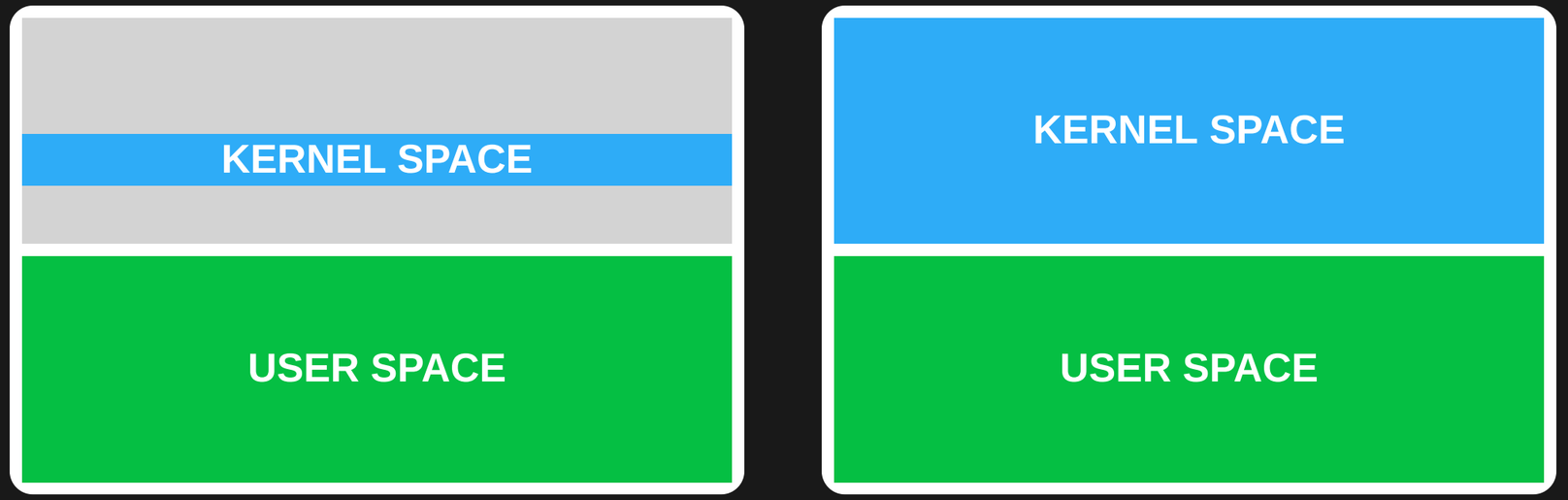
Linux KPTI (Kernel page-table isolation)



for security reasons, the entire kernel is usually not always mapped

MODERN LINUX

Linux KPTI (Kernel page-table isolation)



USER MODE KERNEL MODE

Linux KPTI (Kernel page-table isolation):

- uses separate page table for kernel and user mode
- isolates user space and kernel space memory
- basic measure against Meltdown vulnerability
- degrades performance (5-30%)
- previously known as KAISER
- partial TLB flushes

More about KPTI:

https://gruss.cc/files/kaiser.pdf https://lwn.net/Articles/738975/ https://lwn.net/Articles/741878/ https://en.wikipedia.org/wiki/Kernel_page-table_isolation https://www.kernel.org/doc/html/next/x86/pti.html

Enough theory, let's do some programming!

My setup:

- x86-64 arch
- 6.8.0 kernel

How to inspect memory layout of process on Linux?

pmap

man pmap

```
MAP(1)

NAME
       pmap - report memory map of a process

SYNOPSIS
       pmap [options] pid [...]

DESCRIPTION
       The pmap command reports the memory map of a process or processes.

OPTIONS
       -x, --extended
              Show the extended format.
```

Example 1

```c
#include <stdio.h>

int main() {
  printf("Hello, NDC TECHTOWN!");
  getchar();
  return 0;
}
```

to see memory mappings:

```
./userProgram & pmap $! -XX
```

```
          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
    7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
    7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
    7fff13748000 rw-p 00000000       0  132  16        16  [stack]
    7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
    7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
    7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
    7ba3601c7000 rw-p 00000000       0    8   4         4
    7ba3601af000 rw-p 00000000       0   12   8         8
    7ba360005000 rw-p 00000000       0   52  20        20
    7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
    7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
    7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
    7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
    7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
    5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap]
    5ee2b656e000 rw-p 00003000 5251000    4   4         4  userProgram
    5ee2b656d000 r--p 00002000 5251000    4   4         4  userProgram
    5ee2b656c000 r--p 00002000 5251000    4   4         0  userProgram
    5ee2b656b000 r-xp 00001000 5251000    4   4         0  userProgram
    5ee2b656a000 r--p 00000000 5251000    4   4         0  userProgram
                                         ==== === =========
                                         2684 356       100
```

1

```
          Address
ffffffffff600000
    7fff13783000
    7fff1377f000
    7fff13748000
    7ba360201000
    7ba3601ff000
    7ba3601f5000
    7ba3601ca000
    7ba3601c9000
    7ba3601c7000
    7ba3601af000
    7ba360005000
    7ba360003000
    7ba35ffff000
    7ba35ffb0000
    7ba35fe28000
    7ba35fe00000
    5ee2b7ed8000
    5ee2b656e000
    5ee2b656d000
    5ee2b656c000
    5ee2b656b000
    5ee2b656a000
```

1

```
          Address
ffffffffff600000
    7fff13783000
    7fff1377f000
    7fff13748000
    7ba360201000
    7ba3601ff000
    7ba3601f5000
    7ba3601ca000
    7ba3601c9000
    7ba3601c7000
    7ba3601af000
    7ba360005000
    7ba360003000
    7ba35ffff000
    7ba35ffb0000
    7ba35fe28000
    7ba35fe00000
    5ee2b7ed8000
    5ee2b656e000
    5ee2b656d000
    5ee2b656c000
    5ee2b656b000
    5ee2b656a000
```

2

```
          Address
ffffffffff600000
    7ffde8ff6000
    7ffde8ff2000
    7ffde8e9d000
    72450ee3c000
    72450ee3a000
    72450ee30000
    72450ee05000
    72450ee04000
    72450ee02000
    72450ede5000
    72450ec05000
    72450ec03000
    72450ebff000
    72450ebb0000
    72450ea28000
    72450ea00000
    604e6c7a2000
    604e6b93e000
    604e6b93d000
    604e6b93c000
    604e6b93b000
    604e6b93a000
```

1

```
        Address
fffffffff600000
    7fff13783000
    7fff1377f000
    7fff13748000
    7ba360201000
    7ba3601ff000
    7ba3601f5000
    7ba3601ca000
    7ba3601c9000
    7ba3601c7000
    7ba3601af000
    7ba360005000
    7ba360003000
    7ba35ffff000
    7ba35ffb0000
    7ba35fe28000
    7ba35fe00000
    5ee2b7ed8000
    5ee2b656e000
    5ee2b656d000
    5ee2b656c000
    5ee2b656b000
    5ee2b656a000
```

2

```
        Address
fffffffff600000
    7ffde8ff6000
    7ffde8ff2000
    7ffde8e9d000
    72450ee3c000
    72450ee3a000
    72450ee30000
    72450ee05000
    72450ee04000
    72450ee02000
    72450ede5000
    72450ec05000
    72450ec03000
    72450ebff000
    72450ebb0000
    72450ea28000
    72450ea00000
    604e6c7a2000
    604e6b93e000
    604e6b93d000
    604e6b93c000
    604e6b93b000
    604e6b93a000
```

3

```
        Address
fffffffff600000
    7ffcabdfd000
    7ffcabdf9000
    7ffcabdd1000
    7bc88ec4a000
    7bc88ec48000
    7bc88ec3e000
    7bc88ec13000
    7bc88ec12000
    7bc88ec10000
    7bc88ebf3000
    7bc88ea05000
    7bc88ea03000
    7bc88e9ff000
    7bc88e9b0000
    7bc88e828000
    7bc88e800000
    58ccc0f78000
    58ccbff3e000
    58ccbff3d000
    58ccbff3c000
    58ccbff3b000
    58ccbff3a000
```

ADDRESS SPACE LAYOUT RANDOMIZATION (ASLR)

ADDRESS SPACE LAYOUT RANDOMIZATION (ASLR)

Increase security of binaries:

- randomizes position of some regions (program, mmap, stack and VDSO)
- introduces random offsets before start of some segments (heap, stack)

ADDRESS SPACE LAYOUT RANDOMIZATION (ASLR)

Controlled by `/proc/sys/kernel/randomize_va_space` on Linux.

- `randomize_va_space = 0` - disabled
- `randomize_va_space = 1` - conservative randomization
- `randomize_va_space = 2` - full randomization (default)

Binary must be compiled as Position Independent Executable
in order for ALSR to work.

```
          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
    7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
    7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
    7fff13748000 rw-p 00000000       0  132  16        16  [stack]
    7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
    7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
    7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
    7ba3601c7000 rw-p 00000000       0    8   4         4
    7ba3601af000 rw-p 00000000       0   12   8         8
    7ba360005000 rw-p 00000000       0   52  20        20
    7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
    7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
    7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
    7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
    7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
    5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap]
    5ee2b656e000 rw-p 00003000 5251000    4   4         4  userProgram
    5ee2b656d000 r--p 00002000 5251000    4   4         4  userProgram
    5ee2b656c000 r--p 00002000 5251000    4   4         0  userProgram
    5ee2b656b000 r-xp 00001000 5251000    4   4         0  userProgram
    5ee2b656a000 r--p 00000000 5251000    4   4         0  userProgram
                                         ==== === =========
                                         2684 356       100
```

48 BITS ADDRESS?

```
ffffffffff600000
    7fff13783000
```

48 BITS ADDRESS?

```
fffffffff600000
    7fff13783000 = 12 x 4 = 48bits
```

48 BITS ADDRESS?

```
ffffffffff600000 = 16 x 4 = 64bits???
    7fff13783000 = 12 x 4 = 48bits
```

48 BITS ADDRESS?

`ffffffffff600000`

48 BITS ADDRESS?

```
1111111111111111111111111111111111111111011000000000000000000000
```

48 BITS ADDRESS?

1111111111111111111111111111111011000000000000000000000000

48 BITS ADDRESS?

1111111111111111111111111111111011000000000000000000000000

48 BITS ADDRESS?

0000000000000000011111111111111111111101100000000000000000000000

48 BITS ADDRESS?

kernel space:

1111111111111111111111111111111111111111111111111111

-

1111111111111111100000000000000000000000000000000000

user space:

0000000000000000011111111111111111111111111111111111

-

0000000000000000000000000000000000000000000000000000

48 BITS ADDRESS?

kernel space:

FFFFFFFFFFFFFFFF

-

FFFF800000000000

user space:

00007FFFFFFFFFFF

-

0000000000000000

more info

https://en.wikipedia.org/wiki/X86-64#Canonical_form_addresses

```
 1           Address Perm   Offset   Inode Size Rss Anonymous  Mapping
 2 ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
 3     7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
 4     7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
 5     7fff13748000 rw-p 00000000       0  132  16        16  [stack]
 6     7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
 7     7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
 8     7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
 9     7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
10     7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
11     7ba3601c7000 rw-p 00000000       0    8   4         4
12     7ba3601af000 rw-p 00000000       0   12   8         8
13     7ba360005000 rw-p 00000000       0   52  20        20
14     7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
15     7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
16     7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
17     7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
18     7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
19     5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap]
20     5ee2b656e000 rw-p 00003000 5251000    4   4         4  userProgram
21     5ee2b656d000 r--p 00002000 5251000    4   4         4  userProgram
22     5ee2b656c000 r--p 00002000 5251000    4   4         0  userProgram
23     5ee2b656b000 r-xp 00001000 5251000    4   4         0  userProgram
24     5ee2b656a000 r--p 00000000 5251000    4   4         0  userProgram
25                                          ==== === =========
26                                          2684 356       100
```

```
 1           Address Perm   Offset   Inode Size Rss Anonymous  Mapping
 2 ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
 3     7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
 4     7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
 5     7fff13748000 rw-p 00000000       0  132  16        16  [stack]
 6     7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
 7     7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
 8     7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
 9     7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
10     7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
11     7ba3601c7000 rw-p 00000000       0    8   4         4
12     7ba3601af000 rw-p 00000000       0   12   8         8
13     7ba360005000 rw-p 00000000       0   52  20        20
14     7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
15     7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
16     7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
17     7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
18     7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
19     5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap]
20     5ee2b656e000 rw-p 00003000 5251000    4   4         4  userProgram
21     5ee2b656d000 r--p 00002000 5251000    4   4         4  userProgram
22     5ee2b656c000 r--p 00002000 5251000    4   4         0  userProgram
23     5ee2b656b000 r-xp 00001000 5251000    4   4         0  userProgram
24     5ee2b656a000 r--p 00000000 5251000    4   4         0  userProgram
25                                          ==== === =========
26                                          2684 356       100
```

WHAT IS IN THE ELF FILE?

Content of ELF files is structured into **sections**.

To display ELF sections:

```
readelf -S userProgram
```

```
There are 31 section headers, starting at offset 0x3728:
Section Headers:
  [Nr] Name              Type             Address           Off    Size
  [ 0]                   NULL             0000000000000000 000000 000000
  [ 1] .interp           PROGBITS         0000000000000318 000318 00001c
  [ 2] .note.gnu.property NOTE            0000000000000338 000338 000020
  [ 3] .note.gnu.build-id NOTE            0000000000000358 000358 000024
  [ 4] .note.ABI-tag     NOTE             000000000000037c 00037c 000020
  [ 5] .gnu.hash         GNU_HASH         00000000000003a0 0003a0 000024
  [ 6] .dynsym           DYNSYM           00000000000003c8 0003c8 0000c0
  [ 7] .dynstr           STRTAB           0000000000000488 000488 000097
  [ 8] .gnu.version      VERSYM           0000000000000520 000520 000010
  [ 9] .gnu.version_r    VERNEED          0000000000000530 000530 000030
  [10] .rela.dyn         RELA             0000000000000560 000560 0000c0
  [11] .rela.plt         RELA             0000000000000620 000620 000030
  [12] .init             PROGBITS         0000000000001000 001000 00001b
  [13] .plt              PROGBITS         0000000000001020 001020 000030
  [14] .plt.got          PROGBITS         0000000000001050 001050 000008
  [15] .text             PROGBITS         0000000000001060 001060 00011a
  [16] .fini             PROGBITS         000000000000117c 00117c 00000d
  [17] .rodata           PROGBITS         0000000000002000 002000 000019
  [18] .eh_frame_hdr     PROGBITS         000000000000201c 00201c 00002c
  [19] .eh_frame         PROGBITS         0000000000002048 002048 000094
  (...)
```

WHICH SECTIONS ARE MAPPED?

This information of WHAT to load WHERE is stored in **program headers**.

To display program headers:

```
readelf -l userProgram
```

```
Program Headers:
    Type           Offset   FileSiz  MemSiz   Flg Align
    PHDR           0x000040 0x0002d8 0x0002d8 R   0x8
    INTERP         0x000318 0x00001c 0x00001c R   0x1
    LOAD           0x000000 0x000688 0x000688 R   0x1000
    LOAD           0x001000 0x000181 0x000181 R E 0x1000
    LOAD           0x002000 0x0000d4 0x0000d4 R   0x1000
    LOAD           0x002dc8 0x000258 0x000260 RW  0x1000
    DYNAMIC        0x002dd8 0x0001e0 0x0001e0 RW  0x8
    NOTE           0x000338 0x000020 0x000020 R   0x8
    NOTE           0x000358 0x000044 0x000044 R   0x4
    GNU_PROPERTY   0x000338 0x000020 0x000020 R   0x8
    GNU_EH_FRAME   0x00201c 0x00002c 0x00002c R   0x4
    GNU_STACK      0x000000 0x000000 0x000000 RW  0x10
    GNU_RELRO      0x002dc8 0x000238 0x000238 R   0x1

Section to Segment mapping:
    00
    01     .interp
    02     .interp .note.gnu.property .note.gnu.build-id .note.ABI-tag \
           .gnu.hash .dynsym .dynstr .gnu.version .gnu.version_r \
           .rela.dyn .rela.plt
    03     .init .plt .plt.got .text .fini
    04     .rodata .eh_frame_hdr .eh_frame
    05     .init_array .fini_array .dynamic .got .got.plt .data .bss
    06     .dynamic
    07     .note.gnu.property
    08     .note.gnu.build-id .note.ABI-tag
    09     .note.gnu.property
```

```
Program Headers:
  Type           Offset   FileSiz  MemSiz   Flg Align
  PHDR           0x000040 0x0002d8 0x0002d8 R   0x8
  INTERP         0x000318 0x00001c 0x00001c R   0x1
  LOAD           0x000000 0x000688 0x000688 R   0x1000
  LOAD           0x001000 0x000181 0x000181 R E 0x1000
  LOAD           0x002000 0x0000d4 0x0000d4 R   0x1000
  LOAD           0x002dc8 0x000258 0x000260 RW  0x1000
  DYNAMIC        0x002dd8 0x0001e0 0x0001e0 RW  0x8
  NOTE           0x000338 0x000020 0x000020 R   0x8
  NOTE           0x000358 0x000044 0x000044 R   0x4
  GNU_PROPERTY   0x000338 0x000020 0x000020 R   0x8
  GNU_EH_FRAME   0x00201c 0x00002c 0x00002c R   0x4
  GNU_STACK      0x000000 0x000000 0x000000 RW  0x10
  GNU_RELRO      0x002dc8 0x000238 0x000238 R   0x1

Section to Segment mapping:
  00
  01     .interp
  02     .interp .note.gnu.property .note.gnu.build-id .note.ABI-tag \
          .gnu.hash .dynsym .dynstr .gnu.version .gnu.version_r \
          .rela.dyn .rela.plt
  03     .init .plt .plt.got .text .fini
  04     .rodata .eh_frame_hdr .eh_frame
  05     .init_array .fini_array .dynamic .got .got.plt .data .bss
  06     .dynamic
  07     .note.gnu.property
  08     .note.gnu.build-id .note.ABI-tag
  09     .note.gnu.property
```

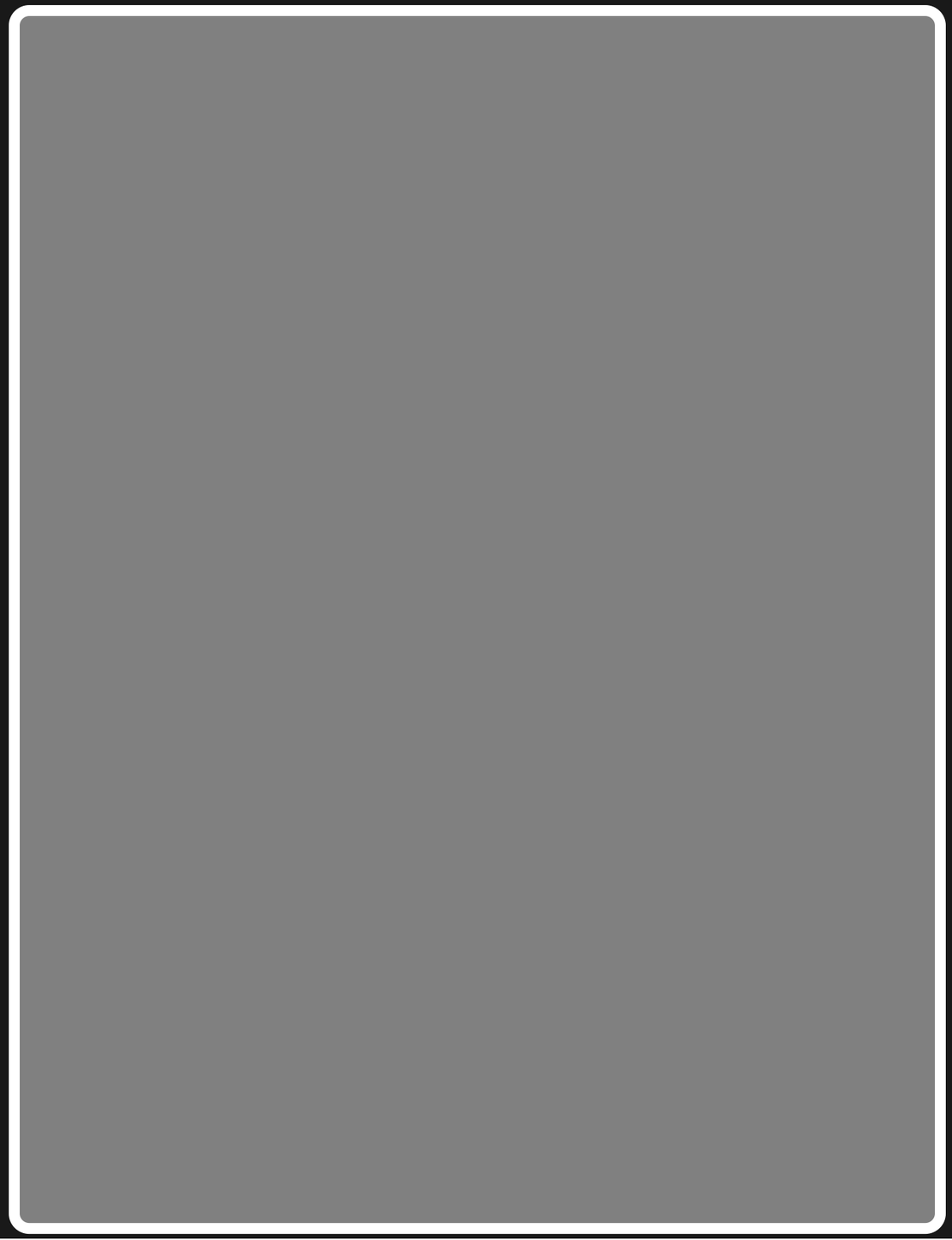
```
 4     INTERP         0x000318 0x00001c 0x00001c R   0x1
 5     LOAD           0x000000 0x000688 0x000688 R   0x1000
 6     LOAD           0x001000 0x000181 0x000181 R E 0x1000
 7     LOAD           0x002000 0x0000d4 0x0000d4 R   0x1000
 8     LOAD           0x002dc8 0x000258 0x000260 RW  0x1000
 9     DYNAMIC        0x002dd8 0x0001e0 0x0001e0 RW  0x8
10     NOTE           0x000338 0x000020 0x000020 R   0x8
11     NOTE           0x000358 0x000044 0x000044 R   0x4
12     GNU_PROPERTY   0x000338 0x000020 0x000020 R   0x8
13     GNU_EH_FRAME   0x00201c 0x00002c 0x00002c R   0x4
14     GNU_STACK      0x000000 0x000000 0x000000 RW  0x10
15     GNU_RELRO      0x002dc8 0x000238 0x000238 R   0x1
16
17 Section to Segment mapping:
18     00
19     01     .interp
20     02     .interp .note.gnu.property .note.gnu.build-id .note.ABI-tag \
21            .gnu.hash .dynsym .dynstr .gnu.version .gnu.version_r \
22            .rela.dyn .rela.plt
23     03     .init .plt .plt.got .text .fini
24     04     .rodata .eh_frame_hdr .eh_frame
25     05     .init_array .fini_array .dynamic .got .got.plt .data .bss
26     06     .dynamic
27     07     .note.gnu.property
28     08     .note.gnu.build-id .note.ABI-tag
29     09     .note.gnu.property
30     10     .eh_frame_hdr
31     11
32     12     .init_array .fini_array .dynamic .got
```

```
          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
    7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
    7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
    7fff13748000 rw-p 00000000       0  132  16        16  [stack]
    7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
    7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
    7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
    7ba3601c7000 rw-p 00000000       0    8   4         4
    7ba3601af000 rw-p 00000000       0   12   8         8
    7ba360005000 rw-p 00000000       0   52  20        20
    7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
    7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
    7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
    7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
    7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
    5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap]
    5ee2b656e000 rw-p 00003000 5251000    4   4         4  userProgram
    5ee2b656d000 r--p 00002000 5251000    4   4         4  userProgram
    5ee2b656c000 r--p 00002000 5251000    4   4         0  userProgram
    5ee2b656b000 r-xp 00001000 5251000    4   4         0  userProgram
    5ee2b656a000 r--p 00000000 5251000    4   4         0  userProgram
                                        ==== === =========
                                        2684 356       100
```

```
 1          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
 2 ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
 3     7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
 4     7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
 5     7fff13748000 rw-p 00000000       0  132  16        16  [stack]
 6     7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
 7     7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
 8     7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
 9     7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
10     7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
11     7ba3601c7000 rw-p 00000000       0    8   4         4
12     7ba3601af000 rw-p 00000000       0   12   8         8
13     7ba360005000 rw-p 00000000       0   52  20        20
14     7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
15     7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
16     7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
17     7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
18     7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
19     5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap]
20     5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
21     5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
22     5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
23     5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
24     5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela.*]
25                                         ==== === =========
26                                         2684 356       100
```

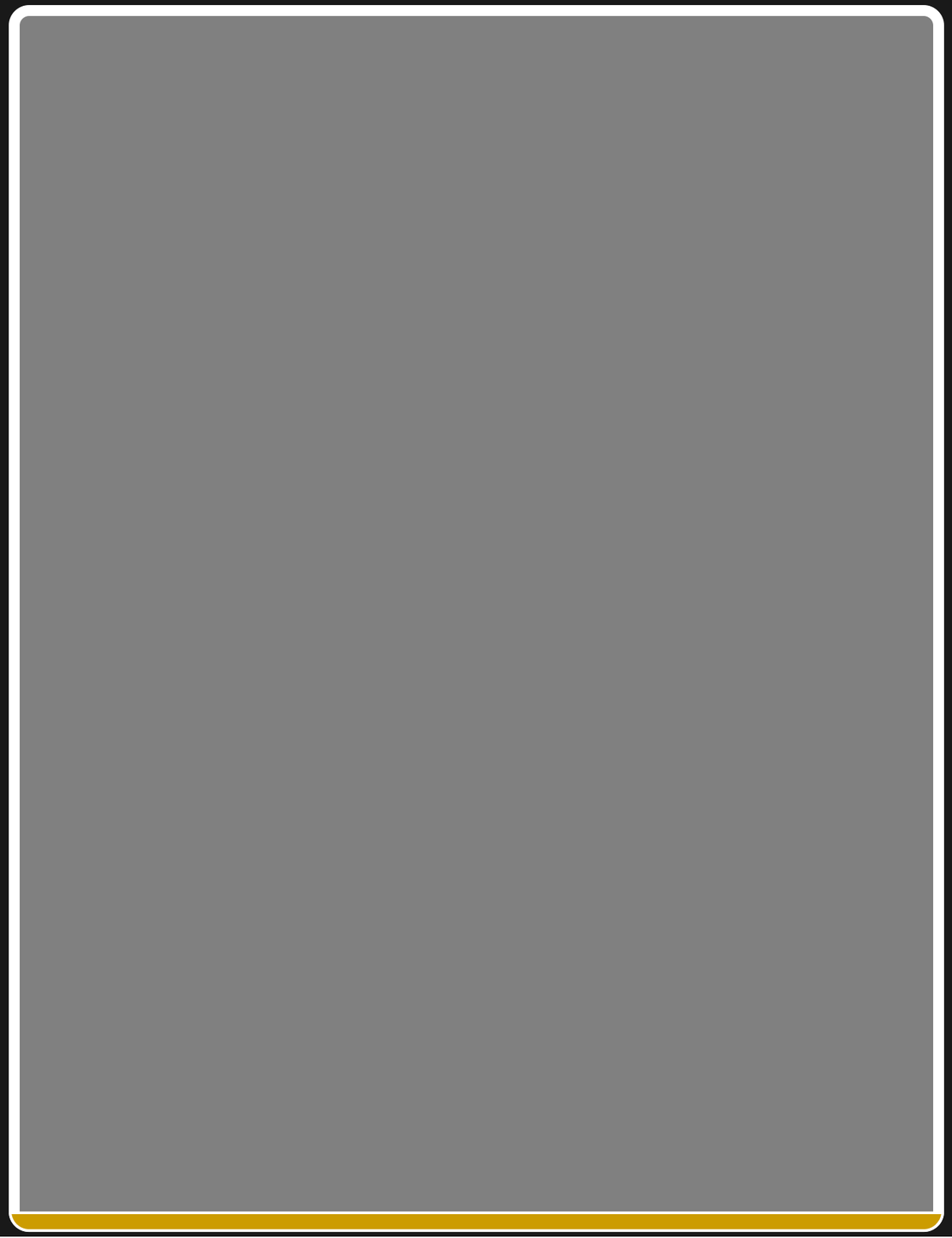
ASLR=1

ASLR=2



ASLR=1

ASLR=2



ASLR=1

ASLR=2


[.got.plt .data .bss]
[.init_array .fini_array .dynamic .got]
[.rodata .eh_frame*]
[.init .plt .plt.got .text .fini]
[.interp .note.* .gnu.* .dyn* .rela*]

```
          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
    7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
    7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
    7fff13748000 rw-p 00000000       0  132  16        16  [stack]
    7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
    7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
    7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
    7ba3601c7000 rw-p 00000000       0    8   4         4
    7ba3601af000 rw-p 00000000       0   12   8         8
    7ba360005000 rw-p 00000000       0   52  20        20
    7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
    7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
    7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
    7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
    7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
    5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap]
    5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
    5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
    5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
    5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
    5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela.*]
                                       ==== === =========
                                       2684 356       100
```

```
          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
    7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
    7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
    7fff13748000 rw-p 00000000       0  132  16        16  [stack]
    7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
    7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
    7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
    7ba3601c7000 rw-p 00000000       0    8   4         4
    7ba3601af000 rw-p 00000000       0   12   8         8
    7ba360005000 rw-p 00000000       0   52  20        20
    7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
    7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
    7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
    7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
    7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
    5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap]
    5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
    5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
    5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
    5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
    5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela.*]
                                       ==== === =========
                                       2684 356       100
```

```c
#include <stdio.h>
#include <stlib.h>
#include <string.h>

int main() {
    const int length = 1*1024;
    char *bytes = malloc(length);
    memset(bytes, 0, length);

    getchar();
 }
```

```c
#include <stdio.h>
#include <stlib.h>
#include <string.h>

int main() {
    const int length = 1*1024;
    char *bytes = malloc(length);
    memset(bytes, 0, length);

    getchar();
 }
```

```
Size Rss Anon  Mapping
   4   0   0  [vsyscall]
   8   4   0  [vdso]
  16   0   0  [vvar]
 132  16  16  [stack]
   8   8   8  ld-linux-x86-64.so.2
   8   8   8  ld-linux-x86-64.so.2
  40  40   0  ld-linux-x86-64.so.2
 172 172   0  ld-linux-x86-64.so.2
   4   4   0  ld-linux-x86-64.so.2
   8   4   4
  12   8   8
  52  20  20
   8   8   8  libc.so.6
  16  16  16  libc.so.6
 316   0   0  libc.so.6
1568 864   0  libc.so.6
 160 160   0  libc.so.6
 132   4   4  [heap]
   4   4   4  [.got.plt .data .bss]
   4   4   4  [.*_array .dynamic .got]
   4   4   0  [.rodata .eh_frame*]
   4   4   0  [.init .plt* .text .fini]
   4   4   0  [.interp .note.* .gnu.*
                    .dyn* .rela.*]
```

```c
#include <stdio.h>
#include <stlib.h>
#include <string.h>

int main() {
    const int length = 10*1024;
    char *bytes = malloc(length);
    memset(bytes, 0, length);

    getchar();
 }
```

```
Size Rss Anon  Mapping
   4   0   0  [vsyscall]
   8   4   0  [vdso]
  16   0   0  [vvar]
 132  16  16  [stack]
   8   8   8  ld-linux-x86-64.so.2
   8   8   8  ld-linux-x86-64.so.2
  40  40   0  ld-linux-x86-64.so.2
 172 172   0  ld-linux-x86-64.so.2
   4   4   0  ld-linux-x86-64.so.2
   8   4   4
  12   8   8
  52  20  20
   8   8   8  libc.so.6
  16  16  16  libc.so.6
 316   0   0  libc.so.6
1568 864   0  libc.so.6
 160 160   0  libc.so.6
 132  16  16  [heap]
   4   4   4  [.got.plt .data .bss]
   4   4   4  [.*_array .dynamic .got]
   4   4   0  [.rodata .eh_frame*]
   4   4   0  [.init .plt* .text .fini]
   4   4   0  [.interp .note.* .gnu.*
                    .dyn* .rela.*]
```

```c
#include <stdio.h>
#include <stlib.h>
#include <string.h>

int main() {
    const int length = 100*1024;
    char *bytes = malloc(length);
    memset(bytes, 0, length);

    getchar();
 }
```

```
Size Rss Anon  Mapping
   4   0   0  [vsyscall]
   8   4   0  [vdso]
  16   0   0  [vvar]
 132  16  16  [stack]
   8   8   8  ld-linux-x86-64.so.2
   8   8   8  ld-linux-x86-64.so.2
  40  40   0  ld-linux-x86-64.so.2
 172 172   0  ld-linux-x86-64.so.2
   4   4   0  ld-linux-x86-64.so.2
   8   4   4
  12   8   8
  52  20  20
   8   8   8  libc.so.6
  16  16  16  libc.so.6
 316   0   0  libc.so.6
1568 864   0  libc.so.6
 160 160   0  libc.so.6
 132 104 104  [heap]
   4   4   4  [.got.plt .data .bss]
   4   4   4  [.*_array .dynamic .got]
   4   4   0  [.rodata .eh_frame*]
   4   4   0  [.init .plt* .text .fini]
   4   4   0  [.interp .note.* .gnu.*
                    .dyn* .rela.*]
```

```c
#include <stdio.h>
#include <stlib.h>
#include <string.h>

int main() {
    const int length = 120*1024;
    char *bytes = malloc(length);
    memset(bytes, 0, length);

    getchar();
 }
```

```
Size Rss Anon  Mapping
   4   0   0  [vsyscall]
   8   4   0  [vdso]
  16   0   0  [vvar]
 132  16  16  [stack]
   8   8   8  ld-linux-x86-64.so.2
   8   8   8  ld-linux-x86-64.so.2
  40  40   0  ld-linux-x86-64.so.2
 172 172   0  ld-linux-x86-64.so.2
   4   4   0  ld-linux-x86-64.so.2
   8   4   4
  12   8   8
  52  20  20
   8   8   8  libc.so.6
  16  16  16  libc.so.6
 316   0   0  libc.so.6
1568 864   0  libc.so.6
 160 160   0  libc.so.6
 132 124 124  [heap]
   4   4   4  [.got.plt .data .bss]
   4   4   4  [.*_array .dynamic .got]
   4   4   0  [.rodata .eh_frame*]
   4   4   0  [.init .plt* .text .fini]
   4   4   0  [.interp .note.* .gnu.*
                    .dyn* .rela.*]
```

```c
#include <stdio.h>
#include <stlib.h>
#include <string.h>

int main() {
    const int length = 140*1024;
    char *bytes = malloc(length);
    memset(bytes, 0, length);

    getchar();
 }
```

```
Size Rss Anon  Mapping
   4   0   0  [vsyscall]
   8   4   0  [vdso]
  16   0   0  [vvar]
 132  16  16  [stack]
   8   8   8  ld-linux-x86-64.so.2
   8   8   8  ld-linux-x86-64.so.2
  40  40   0  ld-linux-x86-64.so.2
 172 172   0  ld-linux-x86-64.so.2
   4   4   0  ld-linux-x86-64.so.2
   8   4   4
 156 152 152
  52  20  20
   8   8   8  libc.so.6
  16  16  16  libc.so.6
 316   0   0  libc.so.6
1568 864   0  libc.so.6
 160 160   0  libc.so.6
 132   4   4  [heap]
   4   4   4  [.got.plt .data .bss]
   4   4   4  [.*_array .dynamic .got]
   4   4   0  [.rodata .eh_frame*]
   4   4   0  [.init .plt* .text .fini]
   4   4   0  [.interp .note.* .gnu.*
                    .dyn* .rela.*]
```

WHY IS THIS HAPPENING?

malloc() has 2 ways to allocate memory:

- use segment data [heap]
- use anonymous mmap()

the choice between both methods usually depends on M_MMAP_THRESHOLD parameter

the choice between both methods usually depends on M_MMAP_THRESHOLD parameter

more info: https://man7.org/linux/man-pages/man3/mallopt.3.html

```c
#include <stdio.h>
#include <stlib.h>
#include <string.h>

int main() {
    const int length = 140*1024;
    char *bytes = malloc(length);
    memset(bytes, 0, length);

    getchar();
 }
```

```c
#include <stdio.h>
#include <stlib.h>
#include <string.h>

int main() {
    const int length = 140*1024;
    char *bytes = malloc(length);
    memset(bytes, 0, length);

    getchar();
 }
```

->

```c
#include <stdio.h>
#include <stlib.h>
#include <string.h>
#include <malloc.h>

int main() {
    mallopt(M_MMAP_THRESHOLD,
            200000);
    const int length = 140*1024;
    char *bytes = malloc(length);
    memset(bytes, 0, length);

    getchar();
 }
```

```
 1 Size Rss Anon  Mapping
 2    4   0   0  [vsyscall]
 3    8   4   0  [vdso]
 4   16   0   0  [vvar]
 5  132  16  16  [stack]
 6    8   8   8  ld-linux-x86-64.so.2
 7    8   8   8  ld-linux-x86-64.so.2
 8   40  40   0  ld-linux-x86-64.so.2
 9  172 172   0  ld-linux-x86-64.so.2
10    4   4   0  ld-linux-x86-64.so.2
11    8   4   4
12  156 152 152
13   52  20  20
14    8   8   8  libc.so.6
15   16  16  16  libc.so.6
16  316   0   0  libc.so.6
17 1568 864   0  libc.so.6
18  160 160   0  libc.so.6
19  132   4   4  [heap]
20    4   4   4  [.got.plt .data .bss]
21    4   4   4  [.*_array .dynamic .got]
22    4   4   0  [.rodata .eh_frame*]
23    4   4   0  [.init .plt* .text .fini]
24    4   4   0  [.interp .note.* .gnu.*
25                   .dyn* .rela.*]
```

```
 1 Size Rss Anon  Mapping
 2    4   0   0  [vsyscall]
 3    8   4   0  [vdso]
 4   16   0   0  [vvar]
 5  132  16  16  [stack]
 6    8   8   8  ld-linux-x86-64.so.2
 7    8   8   8  ld-linux-x86-64.so.2
 8   40  40   0  ld-linux-x86-64.so.2
 9  172 172   0  ld-linux-x86-64.so.2
10    4   4   0  ld-linux-x86-64.so.2
11    8   4   4
12  156 152 152
13   52  20  20
14    8   8   8  libc.so.6
15   16  16  16  libc.so.6
16  316   0   0  libc.so.6
17 1568 864   0  libc.so.6
18  160 160   0  libc.so.6
19  132   4   4  [heap]
20    4   4   4  [.got.plt .data .bss]
21    4   4   4  [.*_array .dynamic .got]
22    4   4   0  [.rodata .eh_frame*]
23    4   4   0  [.init .plt* .text .fini]
24    4   4   0  [.interp .note.* .gnu.*
25                     .dyn* .rela.*]
```

->

```
 1 Size Rss Anon  Mapping
 2    4   0   0  [vsyscall]
 3    8   4   0  [vdso]
 4   16   0   0  [vvar]
 5  132  16  16  [stack]
 6    8   8   8  ld-linux-x86-64.so.2
 7    8   8   8  ld-linux-x86-64.so.2
 8   40  40   0  ld-linux-x86-64.so.2
 9  172 172   0  ld-linux-x86-64.so.2
10    4   4   0  ld-linux-x86-64.so.2
11    8   4   4
12   12   8   8
13   52  20  20
14    8   8   8  libc.so.6
15   16  16  16  libc.so.6
16  316   0   0  libc.so.6
17 1568 864   0  libc.so.6
18  160 160   0  libc.so.6
19  272 144 144  [heap]
20    4   4   4  [.got.plt .data .bss]
21    4   4   4  [.*_array .dynamic .got]
22    4   4   0  [.rodata .eh_frame*]
23    4   4   0  [.init .plt* .text .fini]
24    4   4   0  [.interp .note.* .gnu.*
25                     .dyn* .rela.*]
```

takeaway: what we call "heap" is not really a single region of memory

```
 1           Address Perm   Offset   Inode Size Rss Anonymous  Mapping
 2 ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
 3     7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
 4     7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
 5     7fff13748000 rw-p 00000000       0  132  16        16  [stack]
 6     7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
 7     7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
 8     7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
 9     7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
10     7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
11     7ba3601c7000 rw-p 00000000       0    8   4         4
12     7ba3601af000 rw-p 00000000       0   12   8         8
13     7ba360005000 rw-p 00000000       0   52  20        20
14     7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
15     7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
16     7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
17     7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
18     7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
19     5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap]
20     5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
21     5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
22     5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
23     5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
24     5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela.*]
25                                         ==== === =========
26                                         2684 356       100
```

```
 1          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
 2 ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
 3     7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
 4     7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
 5     7fff13748000 rw-p 00000000       0  132  16        16  [stack]
 6     7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
 7     7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
 8     7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
 9     7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
10     7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
11     7ba3601c7000 rw-p 00000000       0    8   4         4
12     7ba3601af000 rw-p 00000000       0   12   8         8
13     7ba360005000 rw-p 00000000       0   52  20        20
14     7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
15     7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
16     7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
17     7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
18     7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
19     5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap] (not really)
20     5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
21     5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
22     5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
23     5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
24     5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela.*]
25                                         ==== === =========
26                                         2684 356       100
```

ASLR=1

ASLR=2


[.got.plt .data .bss]
[.init_array .fini_array .dynamic .got]
[.rodata .eh_frame*]
[.init .plt .plt.got .text .fini]
[.interp .note.* .gnu.* .dyn* .rela*]

ASLR=1

ASLR=2


[.got.plt .data .bss]
[.init_array .fini_array .dynamic .got]
[.rodata .eh_frame*]
[.init .plt .plt.got .text .fini]
[.interp .note.* .gnu.* .dyn* .rela*]

ASLR=1

ASLR=2


[heap] (not really)
[.got.plt .data .bss]
[.init_array .fini_array .dynamic .got]
[.rodata .eh_frame*]
[.init .plt .plt.got .text .fini]
[.interp .note.* .gnu.* .dyn* .rela*]

```
 1          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
 2 ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
 3     7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
 4     7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
 5     7fff13748000 rw-p 00000000       0  132  16        16  [stack]
 6     7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
 7     7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
 8     7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
 9     7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
10     7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
11     7ba3601c7000 rw-p 00000000       0    8   4         4
12     7ba3601af000 rw-p 00000000       0   12   8         8
13     7ba360005000 rw-p 00000000       0   52  20        20
14     7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
15     7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
16     7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
17     7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
18     7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
19     5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap] (not really)
20     5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
21     5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
22     5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
23     5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
24     5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela.*]
25                                         ==== === =========
26                                         2684 356       100
```

```
          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
    7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
    7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
    7fff13748000 rw-p 00000000       0  132  16        16  [stack]
    7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
    7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
    7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
    7ba3601c7000 rw-p 00000000       0    8   4         4
    7ba3601af000 rw-p 00000000       0   12   8         8
    7ba360005000 rw-p 00000000       0   52  20        20
    7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
    7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
    7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
    7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
    7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
    5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap] (not really)
    5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
    5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
    5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
    5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
    5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela.*]
                                       ==== === =========
                                       2684 356       100
```

Problem: syscalls are slow as they require switch between kernel and user mode.

Problem: syscalls are slow as they require switch between kernel and user mode.

Solution: don't require switch to kernel mode for some syscall.

vsyscall (Virtual System Call):

- kernel maps a fixed memory region directly into user space
- contains fast implementations of certain system calls (e.g. `gettimeofday()`)
- eliminates the need for a mode switch to kernel space

vsyscall problems:

- fixed memory location makes it vulnerable to attack
- limited number of supported syscall
- deprecated in favor of vDSO

vdso (Virtual Dynamically-linked Shared Object):

- kernel maps a small shared library into user space
- supports ASLR
- can accommodate more syscalls than vsyscall

vvar:

- read-only variables for VDSO
- exposes necessary kernel variable to user space

more info:

https://lwn.net/Articles/446528/

https://man7.org/linux/man-pages/man7/vdso.7.html

```
          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
    7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
    7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
    7fff13748000 rw-p 00000000       0  132  16        16  [stack]
    7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
    7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
    7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
    7ba3601c7000 rw-p 00000000       0    8   4         4
    7ba3601af000 rw-p 00000000       0   12   8         8
    7ba360005000 rw-p 00000000       0   52  20        20
    7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
    7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
    7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
    7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
    7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
    5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap] (not really)
    5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
    5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
    5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
    5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
    5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela.*]
                                       ==== === =========
                                       2684 356       100
```

```
          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]-|
    7fff13783000 r-xp 00000000       0    8   4         0  [vdso]     |-accelerate some syscalls
    7fff1377f000 r--p 00000000       0   16   0         0  [vvar]-----|
    7fff13748000 rw-p 00000000       0  132  16        16  [stack]
    7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
    7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
    7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
    7ba3601c7000 rw-p 00000000       0    8   4         4
    7ba3601af000 rw-p 00000000       0   12   8         8
    7ba360005000 rw-p 00000000       0   52  20        20
    7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
    7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
    7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
    7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
    7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
    5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap] (not really)
    5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
    5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
    5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
    5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
    5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela.*]
                                       ==== === =========
                                       2684 356       100
```

ASLR=1

ASLR=2


[heap] (not really)
[.got.plt .data .bss]
[.init_array .fini_array .dynamic .got]
[.rodata .eh_frame*]
[.init .plt .plt.got .text .fini]
[.interp .note.* .gnu.* .dyn* .rela*]

ASLR=1

ASLR=2


vsyscall
0xffffffffff600000
[heap] (not really)
[.got.plt .data .bss]
[.init_array .fini_array .dynamic .got]
[.rodata .eh_frame*]
[.init .plt .plt.got .text .fini]
[.interp .note.* .gnu.* .dyn* .rela*]

ASLR=1

ASLR=2


vsyscall
0xffffffffff600000
[heap] (not really)
[.got.plt .data .bss]
[.init_array .fini_array .dynamic .got]
[.rodata .eh_frame*]
[.init .plt .plt.got .text .fini]
[.interp .note.* .gnu.* .dyn* .rela*]

ASLR=1

ASLR=2


vsyscall
0xffffffffff600000
vdso
vvar
[heap] (not really)
[.got.plt .data .bss]
[.init_array .fini_array .dynamic .got]
[.rodata .eh_frame*]
[.init .plt .plt.got .text .fini]
[.interp .note.* .gnu.* .dyn* .rela*]

```
 1          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
 2 ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]-|
 3     7fff13783000 r-xp 00000000       0    8   4         0  [vdso]     |-accelerate some syscalls
 4     7fff1377f000 r--p 00000000       0   16   0         0  [vvar]-----|
 5     7fff13748000 rw-p 00000000       0  132  16        16  [stack]
 6     7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
 7     7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
 8     7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
 9     7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
10     7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
11     7ba3601c7000 rw-p 00000000       0    8   4         4
12     7ba3601af000 rw-p 00000000       0   12   8         8
13     7ba360005000 rw-p 00000000       0   52  20        20
14     7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
15     7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
16     7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
17     7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
18     7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
19     5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap] (not really)
20     5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
21     5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
22     5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
23     5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
24     5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela.*]
25                                         ==== === =========
26                                         2684 356       100
```

```
1           Address Perm   Offset   Inode Size Rss Anonymous  Mapping
2 ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]-|
3     7fff13783000 r-xp 00000000       0    8   4         0  [vdso]     |-accelerate some syscalls
4     7fff1377f000 r--p 00000000       0   16   0         0  [vvar]-----|
5     7fff13748000 rw-p 00000000       0  132  16        16  [stack]
6     7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
7     7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
8     7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
9     7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
10    7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
11    7ba3601c7000 rw-p 00000000       0    8   4         4
12    7ba3601af000 rw-p 00000000       0   12   8         8
13    7ba360005000 rw-p 00000000       0   52  20        20
14    7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
15    7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
16    7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
17    7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
18    7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
19    5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap] (not really)
20    5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
21    5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
22    5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
23    5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
24    5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela.*]
25                                        ==== === =========
26                                        2684 356       100
```

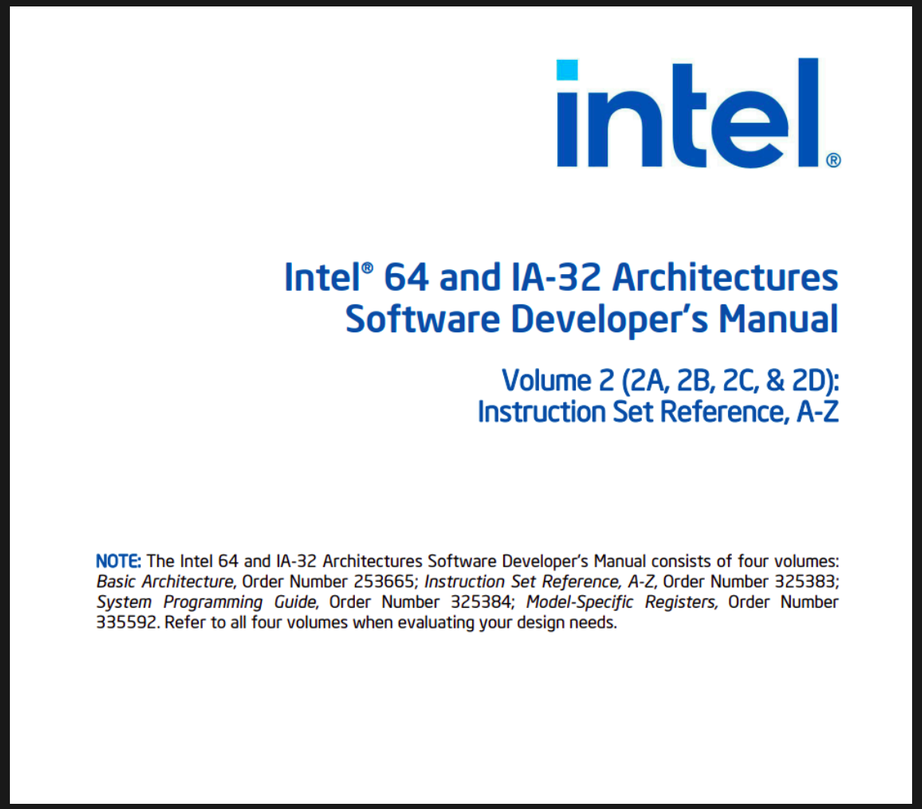
HOW BIG IS THE STACK?

HOW BIG IS THE STACK?

```
$ ulimit -s
8192
```

DOES STACK ALWAYS GROW DOWNWARDS?

DOES STACK ALWAYS GROW DOWNWARDS?



https://cdrdv2-public.intel.com/782156/325383-sdm-vol-2abcd.pdf

PUSH

Description

Decrements the stack pointer and then stores the source operand on the top of the stack. Address and operand sizes are determined and used as follows:

- Address size. The D flag in the current code-segment descriptor determines the default address size; it may be overridden by an instruction prefix (67H).

  The address size is used only when referencing a source operand in memory.

- Operand size. The D flag in the current code-segment descriptor determines the default operand size; it may be overridden by instruction prefixes (66H or REX.W).

  The operand size (16, 32, or 64 bits) determines the amount by which the stack pointer is decremented (2, 4 or 8).

  If the source operand is an immediate of size less than the operand size, a sign-extended value is pushed on the stack. If the source operand is a segment register (16 bits) and the operand size is 64-bits, a zero-extended value is pushed on the stack; if the operand size is 32-bits, either a zero-extended value is pushed on the stack or the segment selector is written on the stack using a 16-bit move. For the last case, all recent Intel Core and Intel Atom processors perform a 16-bit move, leaving the upper portion of the stack location unmodified.

POP

Description

Loads the value from the top of the stack to the location specified with the destination operand (or explicit opcode) and then increments the stack pointer. The destination operand can be a general-purpose register, memory location, or segment register.

Address and operand sizes are determined and used as follows:

- Address size. The D flag in the current code-segment descriptor determines the default address size; it may be overridden by an instruction prefix (67H).

  The address size is used only when writing to a destination operand in memory.

- Operand size. The D flag in the current code-segment descriptor determines the default operand size; it may be overridden by instruction prefixes (66H or REX.W).

  The operand size (16, 32, or 64 bits) determines the amount by which the stack pointer is incremented (2, 4 or 8).

- Stack-address size. Outside of 64-bit mode, the B flag in the current stack-segment descriptor determines the size of the stack pointer (16 or 32 bits); in 64-bit mode, the size of the stack pointer is always 64 bits.

  The stack-address size determines the width of the stack pointer when reading from the stack in memory and when incrementing the stack pointer. (As stated above, the amount by which the stack pointer is incremented is determined by the operand size.)

Stack grows upwards on:

- Hewlett-Packard PA-RISC (HP-PA)
- 8051
- SPARC (in some priviliged modes)

takeaway: stack is both a software and hardware feature

```
          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]-|
    7fff13783000 r-xp 00000000       0    8   4         0  [vdso]     |-accelerate some syscalls
    7fff1377f000 r--p 00000000       0   16   0         0  [vvar]-----|
    7fff13748000 rw-p 00000000       0  132  16        16  [stack]
    7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
    7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
    7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
    7ba3601c7000 rw-p 00000000       0    8   4         4
    7ba3601af000 rw-p 00000000       0   12   8         8
    7ba360005000 rw-p 00000000       0   52  20        20
    7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
    7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
    7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
    7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
    7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
    5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap] (not really)
    5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
    5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
    5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
    5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
    5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela.*]
                                       ==== === =========
                                       2684 356       100
```

Example 4

```
int foo() {
    return 0x55; //BREAKPOINT HERE
}

int bar() {
    return foo();
}

int main(int argc, char *argv[]) {
    long a = 0x9999999999999999; //stored on stack
    int b = bar();
}
```

```
pwndbg> stack

+038    1
+030    0x7fffffffdcb8 —▸ 0x7fffffffe026 ◂— '/home/userProgram'
+028    0x9999999999999999
+020    0
+018    0x555555555179 (main+41)
+010    0x7fffffffdb90 —▸ ...
+008    0x555555555149 (bar+9)
rbp,rsp 0x7fffffffdb60 —▸ ...
```

```
pwndbg> stack

+038    1
+030    0x7fffffffdcb8 —▸ 0x7fffffffe026 ◂— '/home/userProgram'
+028    0x9999999999999999
+020    0
+018    0x555555555179 (main+41)
+010    0x7fffffffdb90 —▸ ...
+008    0x555555555149 (bar+9)
rbp,rsp 0x7fffffffdb60 —▸ ...
```

```
pwndbg> stack

+038    1
+030    0x7fffffffdcb8 —▸ 0x7fffffffe026 ◂— '/home/userProgram'
+028    0x9999999999999999
+020    0
+018    0x555555555179 (main+41)
+010    0x7fffffffdb90 —▸ ...
+008    0x555555555149 (bar+9)
rbp,rsp 0x7fffffffdb60 —▸ ...
```

```
pwndbg> stack

+038    1
+030    0x7fffffffdcb8 —▸ 0x7fffffffe026 ◂— '/home/userProgram'
+028    0x9999999999999999
+020    0
+018    0x555555555179 (main+41)
+010    0x7fffffffdb90 —▸ ...
+008    0x555555555149 (bar+9)
rbp,rsp 0x7fffffffdb60 —▸ ...
```

Example 5

```c
#include <stdio.h>

int main(int argc, char *argv[])
{
    for (int i = 0; i < argc; i++) {
        printf("Argument %d: %s\n", i, argv[i]);
    }
}
```

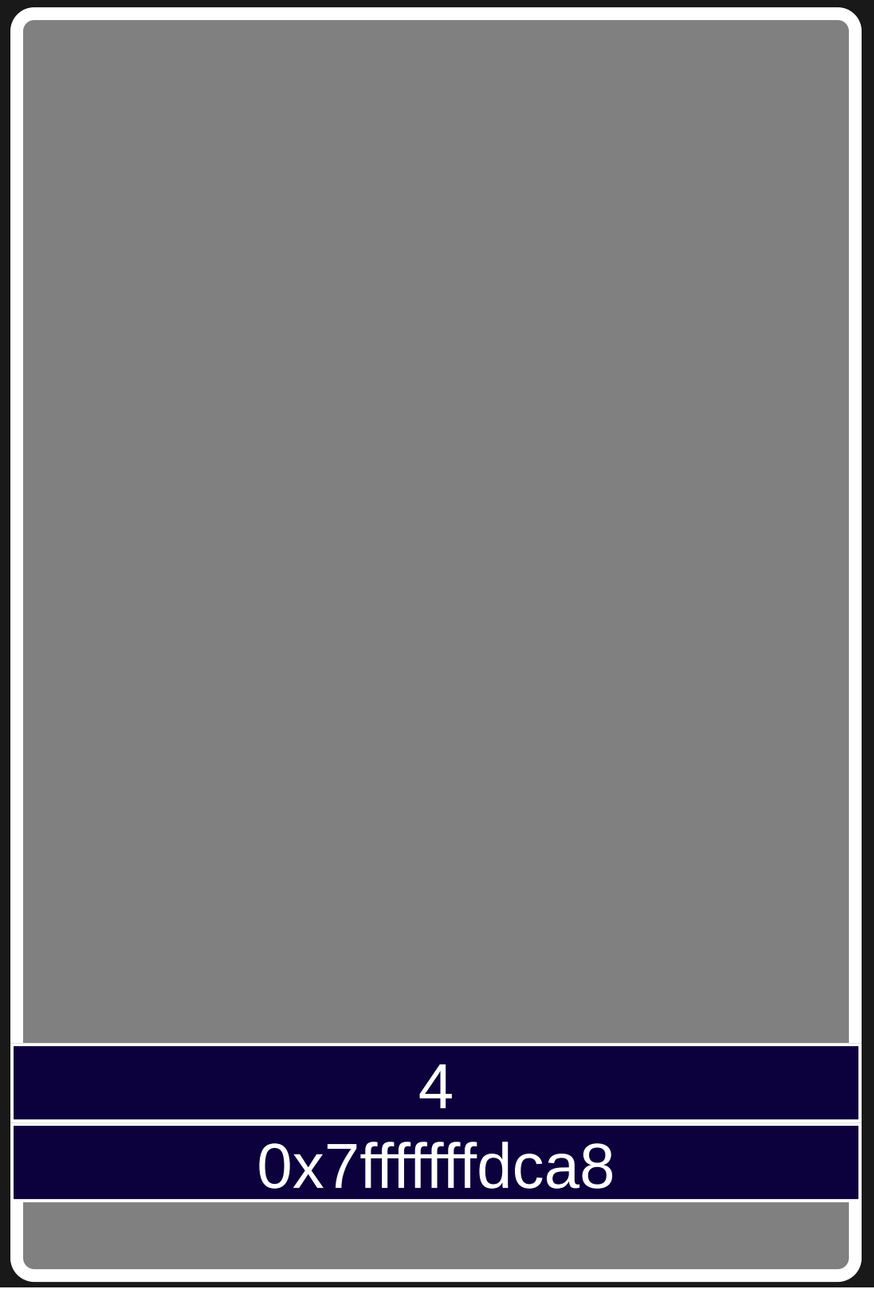
```
./userProgram one two three

Argument 0: ./userProgram
Argument 1: one
Argument 2: two
Argument 3: three
```

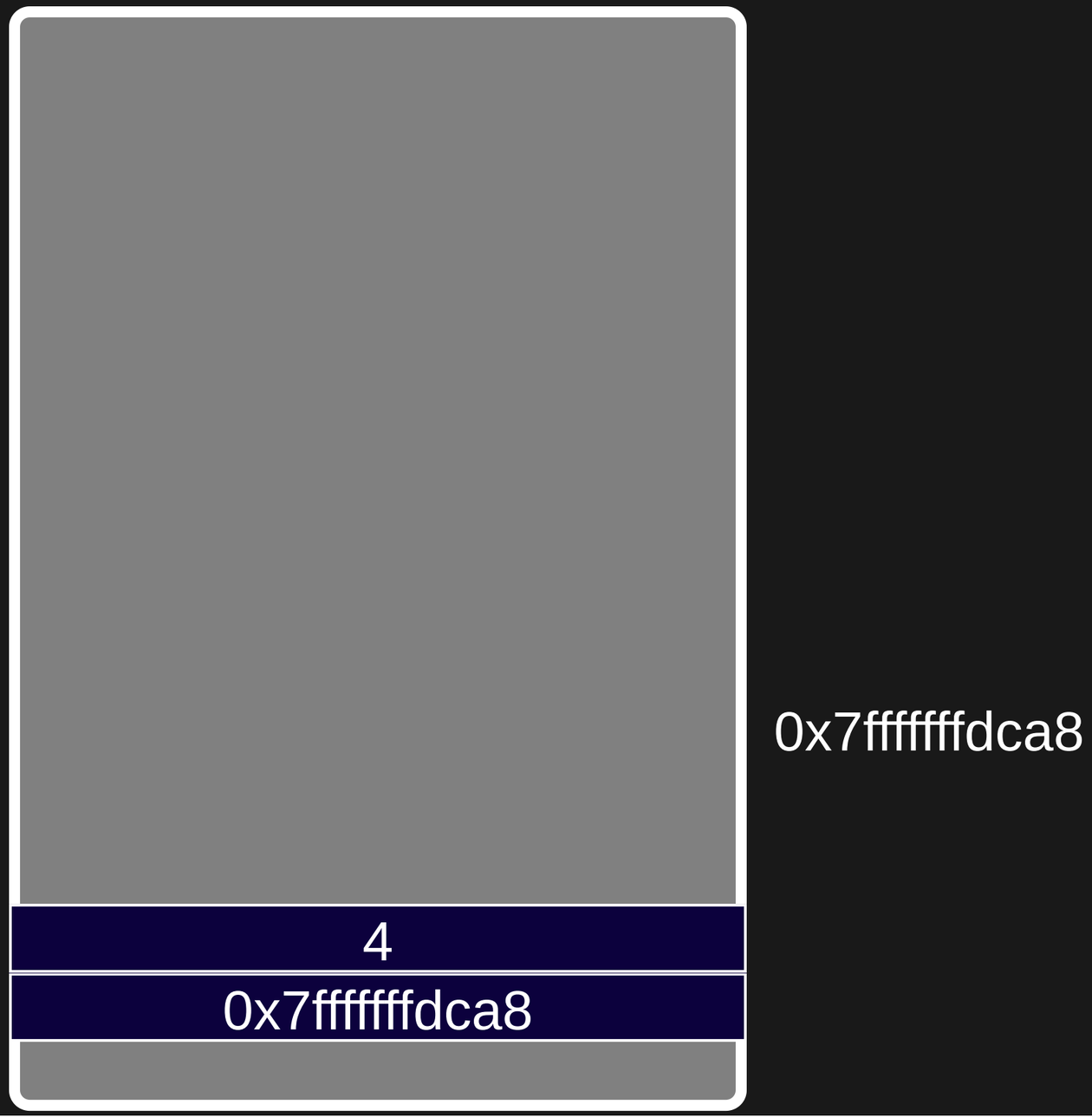
```
pwndbg> stack

 rbp 0x7fffffffdc20 —▸ ...
-008 4
-010 0x7fffffffdca8 —▸ 0x7fffffffe031 ◂— '/home/userProgram'
-018 0x7ffff7fe5af0 (dl_main)
 rsp 0
```

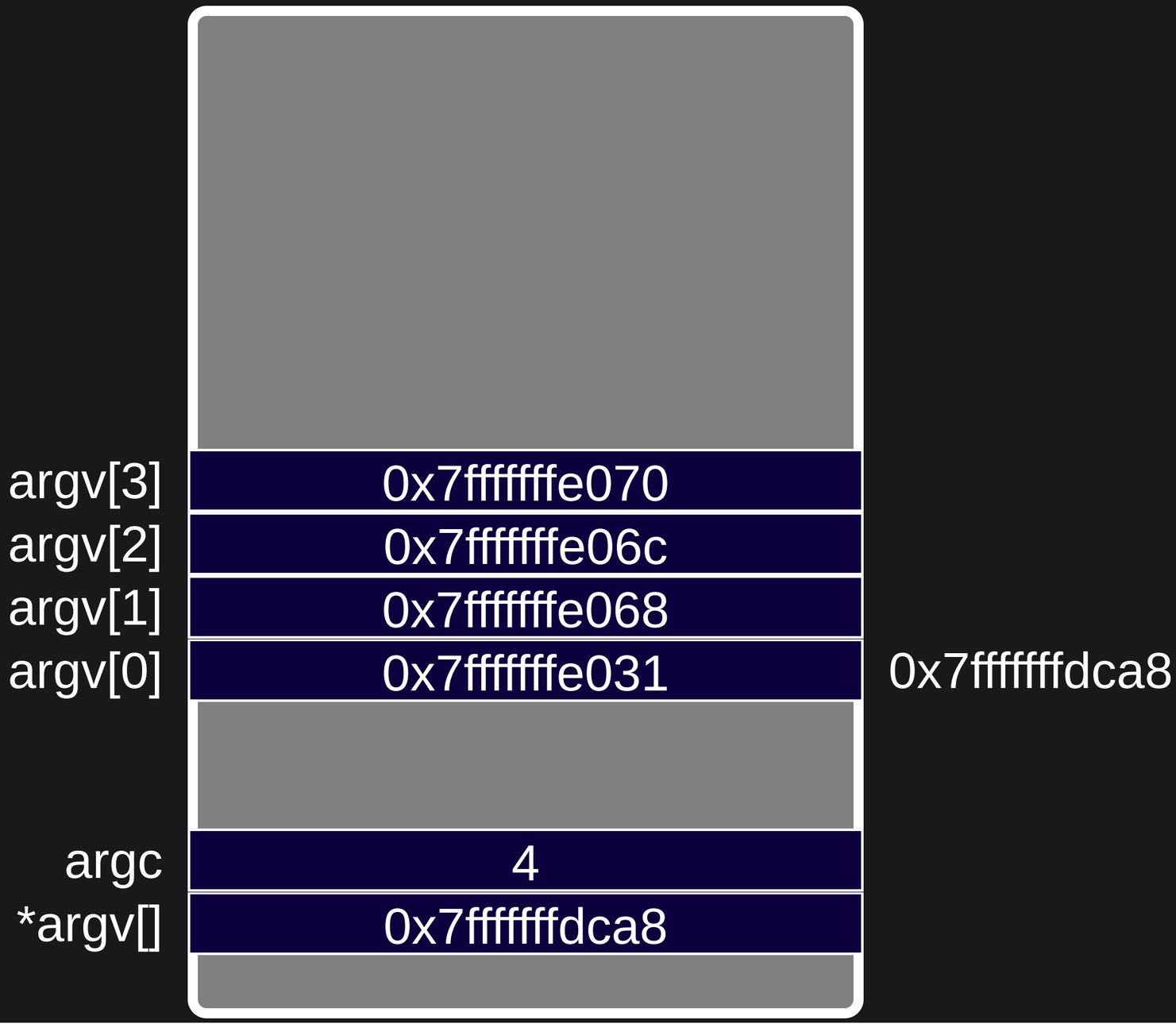
STACK



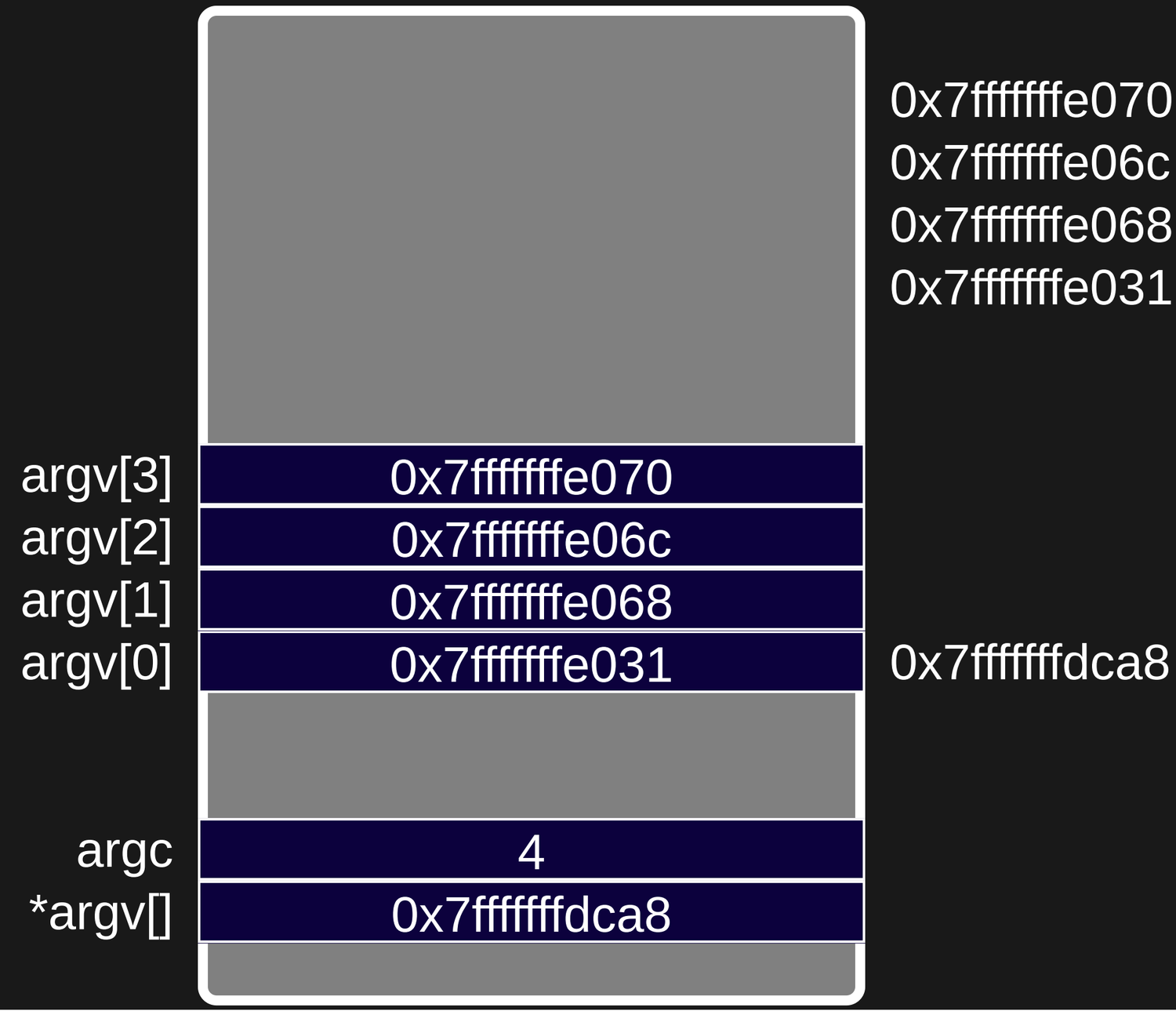
STACK



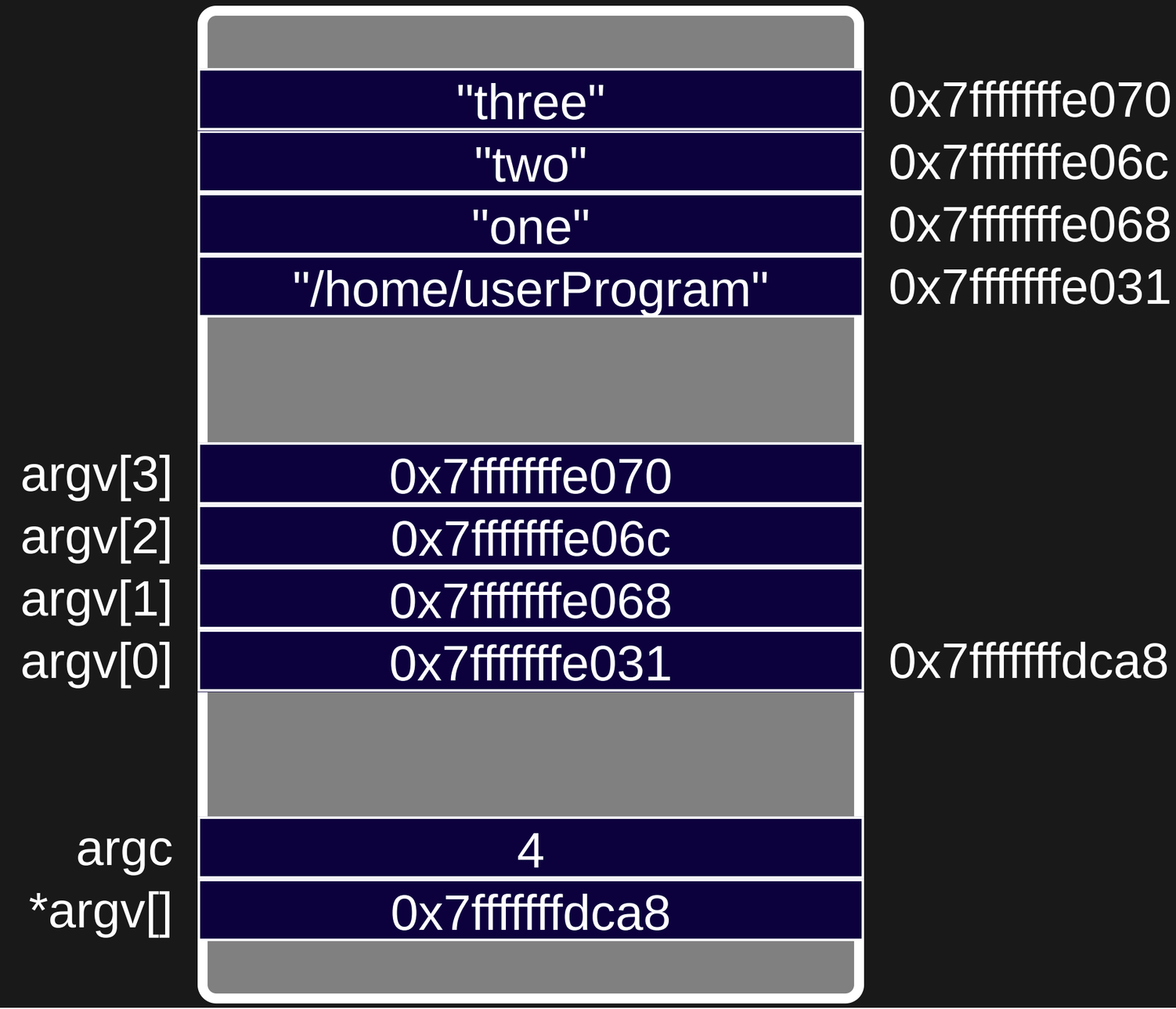
STACK


argv[3]
0x7fffffffe070
argv[2]
0x7fffffffe06c
argv[1]
0x7fffffffe068
argv[0]
0x7fffffffe031
0x7fffffffdca8
argc
4
*argv[]
0x7fffffffdca8

STACK


0x7fffffffe070
0x7fffffffe06c
0x7fffffffe068
0x7fffffffe031
argv[3]
0x7fffffffe070
argv[2]
0x7fffffffe06c
argv[1]
0x7fffffffe068
argv[0]
0x7fffffffe031
0x7fffffffdca8
argc
4
*argv[]
0x7fffffffdca8

STACK


"three"
"two"
"one"
"/home/userProgram"
argv[3]
0x7fffffffe070
argv[2]
0x7fffffffe06c
argv[1]
0x7fffffffe068
argv[0]
0x7fffffffe031
argc
4
*argv[]
0x7fffffffdca8
0x7fffffffe070
0x7fffffffe06c
0x7fffffffe068
0x7fffffffe031
0x7fffffffdca8

STACK

STACK

Example 6

```c
#include <stdio.h>

int main(int argc, char *argv[]) {
    char *ptr = argv[0];

    for (int i = 0; i < 20; i++) {
        printf("%s\n", ptr);
        while (*ptr++ != '\0');
    }
}
```

```
./userProgram one two three

userProgram
one
two
three
SYSTEMD_EXEC_PID=2714
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
SESSION_MANAGER=local/Ubunciak:@/tmp/.ICE-unix/2544,unix/Ubunciak:/tmp/.ICE-uni
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/859d2947_eb5d_476c_800e_a8cbf4
LANG=en_US.UTF-8
XDG_CURRENT_DESKTOP=ubuntu:GNOME
PWD=/home/piotr/repos/DemystifingCode/example6
QT_IM_MODULE=ibus
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
DESKTOP_SESSION=ubuntu
USER=piotr
XDG_MENU_PREFIX=gnome-
HOME=/home/piotr
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
```

STACK

STACK

Example 7 (Unix specific)

```c
#include <stdio.h>

int main(int argc, char *argv[], char *envp[]) {
    for (char **env = envp; *env != NULL; ++env) {
        printf("%s\n", *env);
    }
}
```

STACK

STACK

STACK

STACK

Example 8

```c
#include <stdio.h>
#include <stdint.h>

typedef struct {
    uint64_t type;
    uint64_t value;
} THING;

int main(int argc, char *argv[], char *envp[]) {
    while (*envp++ != NULL);

    THING *ptr = (THING *)envp;
    for (; ptr->type != 0; ptr++)
    {
        printf("%.2ld : %lx\n", ptr->type, ptr->value);
    }
}
```

```
./userProgram

21 : 7ffff7fc3000
33 : d30
10 : 178bfbff
06 : 1000
11 : 64
03 : 555555554040
04 : 38
05 : d
07 : 7ffff7fc5000
08 : 0
09 : 555555555050
0b : 3e8
0c : 3e8
0d : 3e8
0e : 3e8
17 : 0
19 : 7fffffffe089
1a : 2
1f : 7fffffffefea
0f : 7fffffffe099
1b : 1c
1c : 20
```

```
./userProgram

21 : 7ffff7fc3000
33 : d30
10 : 178bfbff
06 : 1000
11 : 64
03 : 555555554040
04 : 38
05 : d
07 : 7ffff7fc5000
08 : 0
09 : 555555555050
0b : 3e8
0c : 3e8
0d : 3e8
0e : 3e8
17 : 0
19 : 7fffffffe089
1a : 2
1f : 7fffffffefea
0f : 7fffffffe099
1b : 1c
1c : 20
```

```
./userProgram

21 : 7ffff7fc3000
33 : d30
10 : 178bfbff
06 : 1000
11 : 64
03 : 555555554040
04 : 38
05 : d
07 : 7ffff7fc5000
08 : 0
09 : 555555555050
0b : 3e8
0c : 3e8
0d : 3e8
0e : 3e8
17 : 0
19 : 7fffffffe089
1a : 2
1f : 7fffffffefea
0f : 7fffffffe099
1b : 1c
1c : 20
```

```
LD_SHOW_AUXV=1 ./userProgram

AT_SYSINFO_EHDR:      0x7ffff7fc3000
AT_MINSIGSTKSZ:       3376
AT_HWCAP:             178bfbff
AT_PAGESZ:            4096
AT_CLKTCK:            100
AT_PHDR:              0x555555554040
AT_PHENT:             56
AT_PHNUM:             13
AT_BASE:              0x7ffff7fc5000
AT_FLAGS:             0x0
AT_ENTRY:             0x555555555050
AT_UID:               1000
AT_EUID:              1000
AT_GID:               1000
AT_EGID:              1000
AT_SECURE:            0
AT_RANDOM:            0x7fffffffe089
AT_HWCAP2:            0x2
AT_EXECFN:            ./userProgram
AT_PLATFORM:          x86_64
AT_??? (0x1b): 0x1c
AT_??? (0x1c): 0x20
```

```
$ gdb userProgram
info auxv

33   AT_SYSINFO_EHDR      System-supplied DSO's ELF header 0x7ffff7fc3000
51   AT_MINSIGSTKSZ       Minimum stack size for signal delivery 0xd30
16   AT_HWCAP             Machine-dependent CPU capability hints 0x178bfbff
6    AT_PAGESZ            System page size                  4096
17   AT_CLKTCK            Frequency of times()              100
3    AT_PHDR              Program headers for program       0x555555554040
4    AT_PHENT             Size of program header entry      56
5    AT_PHNUM             Number of program headers         13
7    AT_BASE              Base address of interpreter       0x7ffff7fc5000
8    AT_FLAGS             Flags                             0x0
9    AT_ENTRY             Entry point of program            0x555555555050
11   AT_UID               Real user ID                      1000
12   AT_EUID              Effective user ID                 1000
13   AT_GID               Real group ID                     1000
14   AT_EGID              Effective group ID                1000
23   AT_SECURE            Boolean, was exec setuid-like? 0
25   AT_RANDOM            Address of 16 random bytes        0x7fffffffe019
26   AT_HWCAP2            Extension of AT_HWCAP             0x2
31   AT_EXECFN            File name of executable           0x7fffffffefc1 "/home/
15   AT_PLATFORM          String identifying platform       0x7fffffffe029 "x86_64
27   AT_RSEQ_FEATURE_SIZE rseq supported feature size       28
```

Auxiliary Vector is meant to pass information about program from kernel to user space.

Auxiliary Vector is meant to pass information about program from kernel to user space.

More info:

https://man7.org/linux/man-pages/man3/getauxval.3.html

STACK

STACK


envp strings
argv strings
auxiliary vector
envp[]
argv[]
argc
*argv[]
*envp[]

```
 1           Address Perm   Offset   Inode Size Rss Anonymous  Mapping
 2 ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]-|
 3     7fff13783000 r-xp 00000000       0    8   4         0  [vdso]     |-accelerate some syscalls
 4     7fff1377f000 r--p 00000000       0   16   0         0  [vvar]-----|
 5     7fff13748000 rw-p 00000000       0  132  16        16  [stack]
 6     7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
 7     7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
 8     7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
 9     7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
10     7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
11     7ba3601c7000 rw-p 00000000       0    8   4         4
12     7ba3601af000 rw-p 00000000       0   12   8         8
13     7ba360005000 rw-p 00000000       0   52  20        20
14     7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
15     7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
16     7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
17     7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
18     7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
19     5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap] (not really)
20     5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
21     5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
22     5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
23     5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
24     5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela.*]
25                                       ==== === =========
26                                       2684 356       100
```

```
          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]-|
    7fff13783000 r-xp 00000000       0    8   4         0  [vdso]     |-accelerate some syscalls
    7fff1377f000 r--p 00000000       0   16   0         0  [vvar]-----|
    7fff13748000 rw-p 00000000       0  132  16        16  [stack] (env vars + aux vector)
    7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
    7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
    7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
    7ba3601c7000 rw-p 00000000       0    8   4         4
    7ba3601af000 rw-p 00000000       0   12   8         8
    7ba360005000 rw-p 00000000       0   52  20        20
    7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
    7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
    7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
    7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
    7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
    5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap] (not really)
    5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
    5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
    5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
    5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
    5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela.*]
                                       ==== === =========
                                       2684 356       100
```

ASLR=1

ASLR=2


vsyscall
0xffffffffff600000
vdso
vvar
[heap] (not really)
[.got.plt .data .bss]
[.init_array .fini_array .dynamic .got]
[.rodata .eh_frame*]
[.init .plt .plt.got .text .fini]
[.interp .note.* .gnu.* .dyn* .rela*]

ASLR=1

ASLR=2


vsyscall
0xffffffffff600000
vdso
vvar
[heap] (not really)
[.got.plt .data .bss]
[.init_array .fini_array .dynamic .got]
[.rodata .eh_frame*]
[.init .plt .plt.got .text .fini]
[.interp .note.* .gnu.* .dyn* .rela*]

ASLR=1

ASLR=2


vsyscall
0xffffffffff600000
vdso
vvar
stack
[heap] (not really)
[.got.plt .data .bss]
[.init_array .fini_array .dynamic .got]
[.rodata .eh_frame*]
[.init .plt .plt.got .text .fini]
[.interp .note.* .gnu.* .dyn* .rela*]

```
 1          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
 2 ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]-|
 3     7fff13783000 r-xp 00000000       0    8   4         0  [vdso]     |-accelerate some syscalls
 4     7fff1377f000 r--p 00000000       0   16   0         0  [vvar]-----|
 5     7fff13748000 rw-p 00000000       0  132  16        16  [stack] (env vars + aux vector)
 6     7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
 7     7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
 8     7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
 9     7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
10     7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
11     7ba3601c7000 rw-p 00000000       0    8   4         4
12     7ba3601af000 rw-p 00000000       0   12   8         8
13     7ba360005000 rw-p 00000000       0   52  20        20
14     7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
15     7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
16     7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
17     7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
18     7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
19     5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap] (not really)
20     5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
21     5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
22     5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
23     5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
24     5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela.*]
25                                       ==== === =========
26                                       2684 356       100
```

```
          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]-|
    7fff13783000 r-xp 00000000       0    8   4         0  [vdso]     |-accelerate some syscalls
    7fff1377f000 r--p 00000000       0   16   0         0  [vvar]-----|
    7fff13748000 rw-p 00000000       0  132  16        16  [stack] (env vars + aux vector)
    7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
    7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
    7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
    7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
    7ba3601c7000 rw-p 00000000       0    8   4         4
    7ba3601af000 rw-p 00000000       0   12   8         8
    7ba360005000 rw-p 00000000       0   52  20        20
    7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
    7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
    7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
    7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
    7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
    5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap] (not really)
    5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
    5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
    5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
    5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
    5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela.*]
                                       ==== === =========
                                       2684 356       100
```

```
          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]-|
    7fff13783000 r-xp 00000000       0    8   4         0  [vdso]     |-accelerate some syscalls
    7fff1377f000 r--p 00000000       0   16   0         0  [vvar]-----|
    7fff13748000 rw-p 00000000       0  132  16        16  [stack] (env vars + aux vector)
    7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2 -|
    7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2  |
    7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2  |
    7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2  |
    7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2  |
    7ba3601c7000 rw-p 00000000       0    8   4         4                        |   Memory
    7ba3601af000 rw-p 00000000       0   12   8         8                        |-- Mapping
    7ba360005000 rw-p 00000000       0   52  20        20                        |   Segment
    7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6             |
    7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6             |
    7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6             |
    7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6             |
    7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6 ------------|
    5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap] (not really)
    5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
    5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
    5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
    5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
    5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela.*]
                                       ==== === =========
                                       2684 356       100
```

ASLR=1

ASLR=2


vsyscall
0xffffffffff600000
vdso
vvar
stack
[heap] (not really)
[.got.plt .data .bss]
[.init_array .fini_array .dynamic .got]
[.rodata .eh_frame*]
[.init .plt .plt.got .text .fini]
[.interp .note.* .gnu.* .dyn* .rela*]

ASLR=1

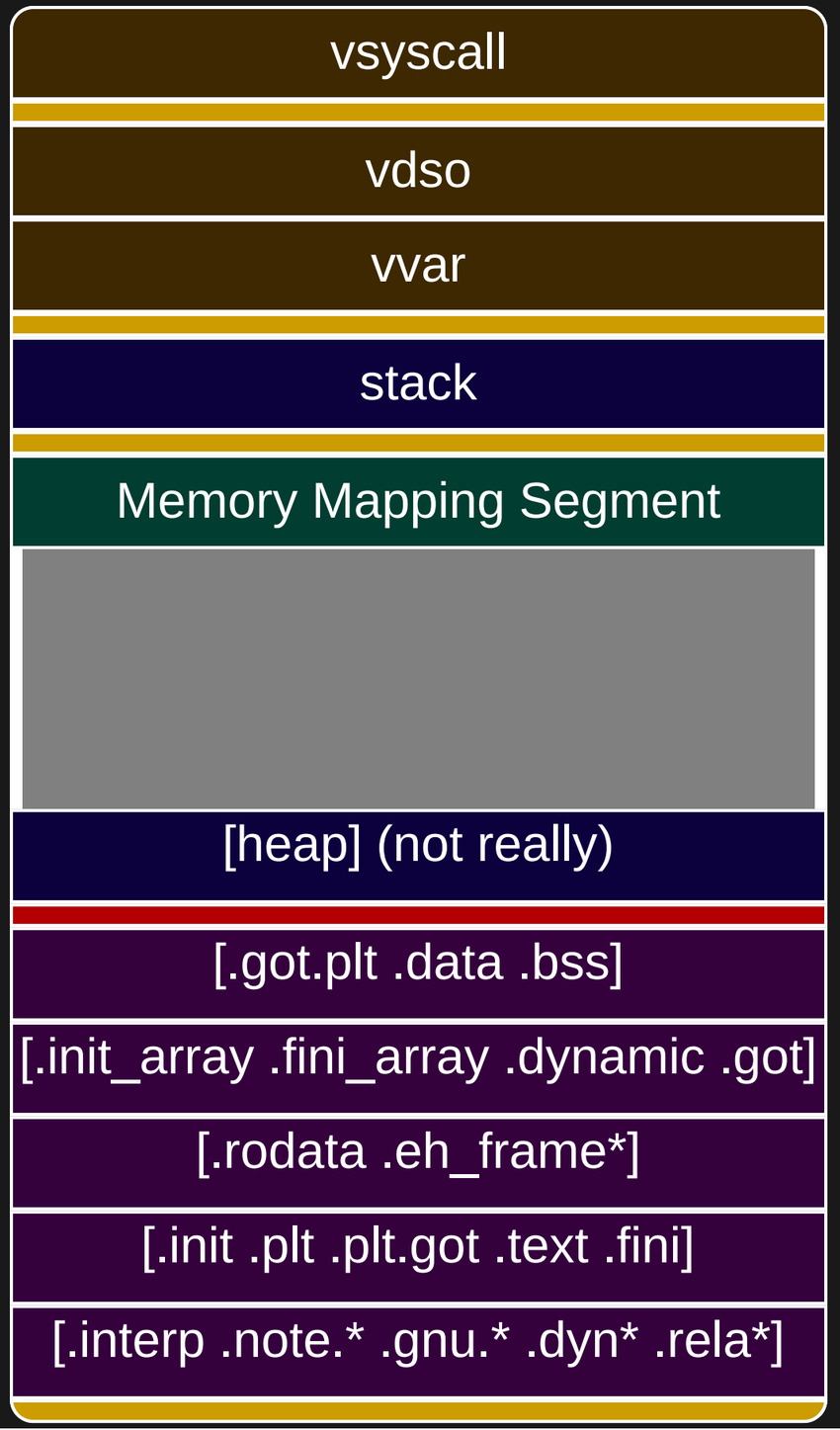
ASLR=2


vsyscall
0xffffffffff600000
vdso
vvar
stack
Memory Mapping Segment
[heap] (not really)
[.got.plt .data .bss]
[.init_array .fini_array .dynamic .got]
[.rodata .eh_frame*]
[.init .plt .plt.got .text .fini]
[.interp .note.* .gnu.* .dyn* .rela*]

let's compare my initial mental model with final result...


STACK
libc
HEAP
???
Qt
CODE


STACK
libc
HEAP
???
Qt
CODE


vsyscall
vdso
vvar
stack
Memory Mapping Segment
[heap] (not really)
[.got.plt .data .bss]
[.init_array .fini_array .dynamic .got]
[.rodata .eh_frame*]
[.init .plt .plt.got .text .fini]
[.interp .note.* .gnu.* .dyn* .rela*]

Can we minimize the process layout further?

Example 1

```c
#include <stdio.h>

int main() {
    printf("Hello, NDC TECHTOWN!");
    getchar();
    return 0;
}
```

```
          Address Perm   Offset   Inode Size Rs Anonymous  Mapping
ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]-|
    7fff13783000 r-xp 00000000       0    8   4         0  [vdso]     |-accelerate some syscalls
    7fff1377f000 r--p 00000000       0   16   0         0  [vvar]-----|
    7fff13748000 rw-p 00000000       0  132  16        16  [stack] (env vars + aux vector)
    7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2 -|
    7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2  |
    7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2  |
    7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2  |
    7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2  |
    7ba3601c7000 rw-p 00000000       0    8   4         4                        |   Memory
    7ba3601af000 rw-p 00000000       0   12   8         8                        |-- Mapping
    7ba360005000 rw-p 00000000       0   52  20        20                        |   Segment
    7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6             |
    7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6             |
    7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6             |
    7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6             |
    7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6 ------------|
    5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap] (not really)
    5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
    5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
    5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
    5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
    5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela.*]
                                       ==== === =========
                                       2684 356       100
```

```
          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]-|
    7fff13783000 r-xp 00000000       0    8   4         0  [vdso]     |-accelerate some syscalls
    7fff1377f000 r--p 00000000       0   16   0         0  [vvar]-----|
    7fff13748000 rw-p 00000000       0  132  16        16  [stack] (env vars + aux vector)
    7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2 -|
    7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2  |
    7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2  |
    7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2  |
    7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2  |
    7ba3601c7000 rw-p 00000000       0    8   4         4                        |   Memory
    7ba3601af000 rw-p 00000000       0   12   8         8                        |-- Mapping
    7ba360005000 rw-p 00000000       0   52  20        20                        |   Segment
    7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6             |
    7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6             |
    7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6             |
    7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6             |
    7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6 ------------|
    5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap] (not really)
    5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
    5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
    5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
    5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
    5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela.*]
                                       ==== === =========
                                       2684 356       100
```

Example 3

```
1 int _start() {
2     while(1) {;}
3 }
```

```
clang main.c -o userProgram -nostdlib
```

```
 1           Address Perm   Offset   Inode Size Rss Anonymous  Mapping
 2 ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
 3     7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
 4     7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
 5     7fff13748000 rw-p 00000000       0  132  16        16  [stack]
 6     7ba360201000 rw-p 00036000  788905   16  16         8  ld-linux-x86-64.so.2
 7     7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
 8     7ba3601ca000 r-xp 00001000  788905  172 168         0  ld-linux-x86-64.so.2
 9     7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
10     7ba3601c7000 rw-p 00000000       0    8   8         8
11     5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap]
12     5ee2b656d000 r--p 00002000 5251000    4   4         4  [.dynamic]
13     5ee2b656c000 r--p 00002000 5251000    4   4         0  [.eh_frame_hdr .eh_frame]
14     5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.text]
15     5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn*]
16                                         ==== === =========
17                                         2684 356       100
```

```
 1          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
 2 ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
 3     7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
 4     7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
 5     7fff13748000 rw-p 00000000       0  132  16        16  [stack]
 6     7ba360201000 rw-p 00036000  788905   16  16         8  ld-linux-x86-64.so.2
 7     7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
 8     7ba3601ca000 r-xp 00001000  788905  172 168         0  ld-linux-x86-64.so.2
 9     7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
10     7ba3601c7000 rw-p 00000000       0    8   8         8
11     5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap]
12     5ee2b656d000 r--p 00002000 5251000    4   4         4  [.dynamic]
13     5ee2b656c000 r--p 00002000 5251000    4   4         0  [.eh_frame_hdr .eh_frame]
14     5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.text]
15     5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn*]
16                                          ==== === =========
17                                          2684 356       100
```

Example 3

```
1 int _start() {
2     while(1) {;}
3 }
```

```
clang main.c -o userProgram -nostdlib -static
```

```
 1           Address Perm   Offset   Inode Size Rss Anonymous  Mapping
 2 ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
 3     7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
 4     7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
 5     7fff13748000 rw-p 00000000       0  132  16        16  [stack]
 6     5ee2b656c000 r--p 00002000 5251000    4   4         0  [.eh_frame_hdr .eh_frame]
 7     5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.text]
 8     5ee2b656a000 r--p 00000000 5251000    4   4         0  [.note.gnu.build-id]
 9                                          ==== === =========
10                                          2684 356       100
```

Example 3

```
1 int _start() {
2     while(1) {;}
3 }
```

```
clang main.c -o userProgram -nostdlib -static -fno-asynchronous-unwind-table
```

```
1           Address Perm   Offset   Inode Size Rss Anonymous  Mapping
2 ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
3     7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
4     7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
5     7fff13748000 rw-p 00000000       0  132  16        16  [stack]
6     5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.text]
7     5ee2b656a000 r--p 00000000 5251000    4   4         0  [.note.gnu.build-id]
8                                         ==== === =========
9                                         2684 356       100
```

There is one [stack] segment per program.

There is one [stack] segment per program.
What about multithreaded programs?

Example 9

```c
#include <stdio.h>
#include <pthread.h>

void* thread_function(void *arg) {
    printf("Hello from the thread!\n");
    getchar();
    return NULL;
}

int main() {
    pthread_t thread;
    pthread_create(&thread, NULL, thread_function, NULL);
    pthread_join(thread, NULL);

    printf("Hello from the main thread!\n");
}
```

```
          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
    ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
        7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
        7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
        7fff13748000 rw-p 00000000       0  132  16        16  [stack]
        7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
        7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
        7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
        7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
        7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
        7ba3601c7000 rw-p 00000000       0    8   4         4
        7ba3601af000 rw-p 00000000       0   12   8         8
        7ba360005000 rw-p 00000000       0   52  20        20
        7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
        7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
        7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
        7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
        7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
        7ba357201000 rw-p 00000000       0 8192  8         8
        7ba357200000 r--p 00000000       0    4  0         0
        7ba350021000 r--p 00000000       0 65404  0         0
        7ba350000000 rw-p 00000000       0  132  8         8
        5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap]
        5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
        5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
        5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
        5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
        5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rela
```

```
 3          7fff13783000 r-xp 00000000       0    8    4         0  [vdso]
 4          7fff1377f000 r--p 00000000       0   16    0         0  [vvar]
 5          7fff13748000 rw-p 00000000       0  132   16        16  [stack]
 6          7ba360201000 rw-p 00038000  788905    8    8         8  ld-linux-x86-64.so.2
 7          7ba3601ff000 r--p 00036000  788905    8    8         8  ld-linux-x86-64.so.2
 8          7ba3601f5000 r--p 0002c000  788905   40   40         0  ld-linux-x86-64.so.2
 9          7ba3601ca000 r-xp 00001000  788905  172  172         0  ld-linux-x86-64.so.2
10          7ba3601c9000 r--p 00000000  788905    4    4         0  ld-linux-x86-64.so.2
11          7ba3601c7000 rw-p 00000000       0    8    4         4
12          7ba3601af000 rw-p 00000000       0   12    8         8
13          7ba360005000 rw-p 00000000       0   52   20        20
14          7ba360003000 rw-p 00202000  788908    8    8         8  libc.so.6
15          7ba35ffff000 r--p 001fe000  788908   16   16        16  libc.so.6
16          7ba35ffb0000 r--p 001b0000  788908  316    0         0  libc.so.6
17          7ba35fe28000 r-xp 00028000  788908 1568  864         0  libc.so.6
18          7ba35fe00000 r--p 00000000  788908  160  160         0  libc.so.6
19          7ba357201000 rw-p 00000000       0 8192    8         8
20          7ba357200000 r--p 00000000       0    4    0         0
21          7ba350021000 r--p 00000000       0 65404    0         0
22          7ba350000000 rw-p 00000000       0  132    8         8
23          5ee2b7ed8000 rw-p 00000000       0  132    4         4  [heap]
24          5ee2b656e000 rw-p 00003000 5251000    4    4         4  [.got.plt .data .bss]
25          5ee2b656d000 r--p 00002000 5251000    4    4         4  [.*_array .dynamic .got]
26          5ee2b656c000 r--p 00002000 5251000    4    4         0  [.rodata .eh_frame*]
27          5ee2b656b000 r-xp 00001000 5251000    4    4         0  [.init .plt* .text .fini]
28          5ee2b656a000 r--p 00000000 5251000    4    4         0  [.interp .note.* .gnu.* .dyn* .rela
29                                          ==== === =========
30                                          2684 356       100
```

```
 3          7fff13783000 r-xp 00000000       0    8    4          0  [vdso]
 4          7fff1377f000 r--p 00000000       0   16    0          0  [vvar]
 5          7fff13748000 rw-p 00000000       0  132   16         16  [stack]
 6          7ba360201000 rw-p 00038000  788905    8    8          8  ld-linux-x86-64.so.2
 7          7ba3601ff000 r--p 00036000  788905    8    8          8  ld-linux-x86-64.so.2
 8          7ba3601f5000 r--p 0002c000  788905   40   40          0  ld-linux-x86-64.so.2
 9          7ba3601ca000 r-xp 00001000  788905  172  172          0  ld-linux-x86-64.so.2
10          7ba3601c9000 r--p 00000000  788905    4    4          0  ld-linux-x86-64.so.2
11          7ba3601c7000 rw-p 00000000       0    8    4          4
12          7ba3601af000 rw-p 00000000       0   12    8          8
13          7ba360005000 rw-p 00000000       0   52   20         20
14          7ba360003000 rw-p 00202000  788908    8    8          8  libc.so.6
15          7ba35ffff000 r--p 001fe000  788908   16   16         16  libc.so.6
16          7ba35ffb0000 r--p 001b0000  788908  316    0          0  libc.so.6
17          7ba35fe28000 r-xp 00028000  788908 1568  864          0  libc.so.6
18          7ba35fe00000 r--p 00000000  788908  160  160          0  libc.so.6
19          7ba357201000 rw-p 00000000       0 8192    8          8
20          7ba357200000 r--p 00000000       0    4    0          0
21          7ba350021000 r--p 00000000       0 65404    0          0
22          7ba350000000 rw-p 00000000       0  132    8          8
23          5ee2b7ed8000 rw-p 00000000       0  132    4          4  [heap]
24          5ee2b656e000 rw-p 00003000 5251000    4    4          4  [.got.plt .data .bss]
25          5ee2b656d000 r--p 00002000 5251000    4    4          4  [.*_array .dynamic .got]
26          5ee2b656c000 r--p 00002000 5251000    4    4          0  [.rodata .eh_frame*]
27          5ee2b656b000 r-xp 00001000 5251000    4    4          0  [.init .plt* .text .fini]
28          5ee2b656a000 r--p 00000000 5251000    4    4          0  [.interp .note.* .gnu.* .dyn* .rela
29                                          ==== === =========
30                                          2684 356       100
```

```
          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
    ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
        7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
        7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
        7fff13748000 rw-p 00000000       0  132  16        16  [stack]
        7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
        7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
        7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
        7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
        7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
        7ba3601c7000 rw-p 00000000       0    8   4         4
        7ba3601af000 rw-p 00000000       0   12   8         8
        7ba360005000 rw-p 00000000       0   52  20        20
        7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
        7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
        7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
        7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
        7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
        7ba357201000 rw-p 00000000       0  8192  8         8  (thread [stack])
        7ba357200000 r--p 00000000       0     4  0         0  (guard page)
        7ba350021000 r--p 00000000       0 65404  0         0
        7ba350000000 rw-p 00000000       0   132  8         8
        5ee2b7ed8000 rw-p 00000000       0  132   4         4  [heap]
        5ee2b656e000 rw-p 00003000 5251000    4   4         4  [.got.plt .data .bss]
        5ee2b656d000 r--p 00002000 5251000    4   4         4  [.*_array .dynamic .got]
        5ee2b656c000 r--p 00002000 5251000    4   4         0  [.rodata .eh_frame*]
        5ee2b656b000 r-xp 00001000 5251000    4   4         0  [.init .plt* .text .fini]
        5ee2b656a000 r--p 00000000 5251000    4   4         0  [.interp .note.* .gnu.* .dyn* .rel
```

Example 9

```c
#include <stdio.h>
#include <pthread.h>

void* thread_function(void *arg) {
    printf("Hello from the thread!\n");
    getchar();
    return NULL;
}

int main() {
    pthread_t thread;
    pthread_create(&thread, NULL, thread_function, NULL);
    pthread_create(&thread, NULL, thread_function, NULL);
    pthread_create(&thread, NULL, thread_function, NULL);
    pthread_create(&thread, NULL, thread_function, NULL);
    pthread_join(thread, NULL);

    printf("Hello from the main thread!\n");
}
```

```
          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
    ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
        7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
        7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
        7fff13748000 rw-p 00000000       0  132  16        16  [stack]
        7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
        7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
        7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
        7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
        7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
        7ba3601c7000 rw-p 00000000       0    8   4         4
        7ba3601af000 rw-p 00000000       0   12   8         8
        7ba360005000 rw-p 00000000       0   52  20        20
        7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
        7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
        7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
        7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
        7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
        7ba357411000 rw-p 00000000       0  8192  8         8
        7ba357410000 r--p 00000000       0     4  0         0
        7ba357321000 rw-p 00000000       0  8192  8         8
        7ba357320000 r--p 00000000       0     4  0         0
        7ba357241000 rw-p 00000000       0  8192  8         8
        7ba357240000 r--p 00000000       0     4  0         0
        7ba357201000 rw-p 00000000       0  8192  4         4
        7ba357200000 r--p 00000000       0     4  0         0
        7ba350021000 r--p 00000000       0 65404  0         0
        7ba350000000 rw-p 00000000       0   132  8         8
```

```
          Address Perm   Offset   Inode Size Rss Anonymous  Mapping
    ffffffffff600000 --xp 00000000       0    4   0         0  [vsyscall]
        7fff13783000 r-xp 00000000       0    8   4         0  [vdso]
        7fff1377f000 r--p 00000000       0   16   0         0  [vvar]
        7fff13748000 rw-p 00000000       0  132  16        16  [stack]
        7ba360201000 rw-p 00038000  788905    8   8         8  ld-linux-x86-64.so.2
        7ba3601ff000 r--p 00036000  788905    8   8         8  ld-linux-x86-64.so.2
        7ba3601f5000 r--p 0002c000  788905   40  40         0  ld-linux-x86-64.so.2
        7ba3601ca000 r-xp 00001000  788905  172 172         0  ld-linux-x86-64.so.2
        7ba3601c9000 r--p 00000000  788905    4   4         0  ld-linux-x86-64.so.2
        7ba3601c7000 rw-p 00000000       0    8   4         4
        7ba3601af000 rw-p 00000000       0   12   8         8
        7ba360005000 rw-p 00000000       0   52  20        20
        7ba360003000 rw-p 00202000  788908    8   8         8  libc.so.6
        7ba35ffff000 r--p 001fe000  788908   16  16        16  libc.so.6
        7ba35ffb0000 r--p 001b0000  788908  316   0         0  libc.so.6
        7ba35fe28000 r-xp 00028000  788908 1568 864         0  libc.so.6
        7ba35fe00000 r--p 00000000  788908  160 160         0  libc.so.6
        7ba357411000 rw-p 00000000       0  8192  8         8  (thread 4 [stack])
        7ba357410000 r--p 00000000       0     4  0         0
        7ba357321000 rw-p 00000000       0  8192  8         8  (thread 3 [stack])
        7ba357320000 r--p 00000000       0     4  0         0
        7ba357241000 rw-p 00000000       0  8192  8         8  (thread 2 [stack])
        7ba357240000 r--p 00000000       0     4  0         0
        7ba357201000 rw-p 00000000       0  8192  4         4  (thread 1 [stack])
        7ba357200000 r--p 00000000       0     4  0         0
        7ba350021000 r--p 00000000       0 65404  0         0
        7ba350000000 rw-p 00000000       0   132  8         8
```

stacks for threads are just created by mmap()

How JIT compilers can put code into memory and execute it?

Getting machine code for function

```
clang func.c -c -o func
```

```c
int add(int a, int b) {
    return a + b;
}
```

Getting machine code for function

```
clang func.c -c -o func
```

```c
int add(int a, int b) {
    return a + b;
}
```

```
objdump func -dC
```

```
Disassembly of section .text:

0000000000000000 <add>:
    0:  55                      push   %rbp
    1:  48 89 e5                mov    %rsp,%rbp
    4:  89 7d fc                mov    %edi,-0x4(%rbp)
    7:  89 75 f8                mov    %esi,-0x8(%rbp)
    a:  8b 45 fc                mov    -0x4(%rbp),%eax
    d:  03 45 f8                add    -0x8(%rbp),%eax
    10: 5d                      pop    %rbp
    11: c3                      ret
```

objdump func -dC

```
Disassembly of section .text:

0000000000000000 <add>:
    0:  55                      push   %rbp
    1:  48 89 e5                mov    %rsp,%rbp
    4:  89 7d fc                mov    %edi,-0x4(%rbp)
    7:  89 75 f8                mov    %esi,-0x8(%rbp)
    a:  8b 45 fc                mov    -0x4(%rbp),%eax
    d:  03 45 f8                add    -0x8(%rbp),%eax
    10: 5d                      pop    %rbp
    11: c3                      ret
```

objdump func -dC

```
Disassembly of section .text:

0000000000000000 <add>:
    0:  55                      push   %rbp
    1:  48 89 e5                mov    %rsp,%rbp
    4:  89 7d fc                mov    %edi,-0x4(%rbp)
    7:  89 75 f8                mov    %esi,-0x8(%rbp)
    a:  8b 45 fc                mov    -0x4(%rbp),%eax
    d:  03 45 f8                add    -0x8(%rbp),%eax
    10: 5d                      pop    %rbp
    11: c3                      ret
```

my machine code:

```
55 48 89 e5 89 7d fc 89 75 f8 8b 45 fc 03 45 f8 5d c3
```

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.n.h>
#include <sys/mman.h>

int main(int argc, char *argv[]) {
    void *memory = mmap(0, 4096, PROT_READ | PROT_WRITE | PROT_EXEC,
                        MAP_PRIVATE | MAP_ANONYMOUS, -1, 0);

    unsigned char code[] = {
        0x55, 0x48, 0x89, 0xe5, 0x89, 0x7d, 0xfc, 0x89, 0x75,
        0xf8, 0x8b, 0x45, 0xfc, 0x03, 0x45, 0xf8, 0x5d, 0xc3
    };
    memcpy(memory, code, sizeof(code));
    int (*func)(int, int) = memory;

    int arg1 = atoi(argv[1]);
    int arg2 = atoi(argv[2]);
    int res = func(arg1, arg2);
    printf("%d + %d = %d\n", arg1, arg2, res);
}
```

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.n.h>
#include <sys/mman.h>

int main(int argc, char *argv[]) {
    void *memory = mmap(0, 4096, PROT_READ | PROT_WRITE | PROT_EXEC,
                        MAP_PRIVATE | MAP_ANONYMOUS, -1, 0);

    unsigned char code[] = {
        0x55, 0x48, 0x89, 0xe5, 0x89, 0x7d, 0xfc, 0x89, 0x75,
        0xf8, 0x8b, 0x45, 0xfc, 0x03, 0x45, 0xf8, 0x5d, 0xc3
    };
    memcpy(memory, code, sizeof(code));
    int (*func)(int, int) = memory;

    int arg1 = atoi(argv[1]);
    int arg2 = atoi(argv[2]);
    int res = func(arg1, arg2);
    printf("%d + %d = %d\n", arg1, arg2, res);
}
```

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.n.h>
#include <sys/mman.h>

int main(int argc, char *argv[]) {
    void *memory = mmap(0, 4096, PROT_READ | PROT_WRITE | PROT_EXEC,
                        MAP_PRIVATE | MAP_ANONYMOUS, -1, 0);

    unsigned char code[] = {
        0x55, 0x48, 0x89, 0xe5, 0x89, 0x7d, 0xfc, 0x89, 0x75,
        0xf8, 0x8b, 0x45, 0xfc, 0x03, 0x45, 0xf8, 0x5d, 0xc3
    };
    memcpy(memory, code, sizeof(code));
    int (*func)(int, int) = memory;

    int arg1 = atoi(argv[1]);
    int arg2 = atoi(argv[2]);
    int res = func(arg1, arg2);
    printf("%d + %d = %d\n", arg1, arg2, res);
}
```

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.n.h>
#include <sys/mman.h>

int main(int argc, char *argv[]) {
    void *memory = mmap(0, 4096, PROT_READ | PROT_WRITE | PROT_EXEC,
                        MAP_PRIVATE | MAP_ANONYMOUS, -1, 0);

    unsigned char code[] = {
        0x55, 0x48, 0x89, 0xe5, 0x89, 0x7d, 0xfc, 0x89, 0x75,
        0xf8, 0x8b, 0x45, 0xfc, 0x03, 0x45, 0xf8, 0x5d, 0xc3
    };
    memcpy(memory, code, sizeof(code));
    int (*func)(int, int) = memory;

    int arg1 = atoi(argv[1]);
    int arg2 = atoi(argv[2]);
    int res = func(arg1, arg2);
    printf("%d + %d = %d\n", arg1, arg2, res);
}
```

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.n.h>
#include <sys/mman.h>

int main(int argc, char *argv[]) {
    void *memory = mmap(0, 4096, PROT_READ | PROT_WRITE | PROT_EXEC,
                        MAP_PRIVATE | MAP_ANONYMOUS, -1, 0);

    unsigned char code[] = {
        0x55, 0x48, 0x89, 0xe5, 0x89, 0x7d, 0xfc, 0x89, 0x75,
        0xf8, 0x8b, 0x45, 0xfc, 0x03, 0x45, 0xf8, 0x5d, 0xc3
    };
    memcpy(memory, code, sizeof(code));
    int (*func)(int, int) = memory;

    int arg1 = atoi(argv[1]);
    int arg2 = atoi(argv[2]);
    int res = func(arg1, arg2);
    printf("%d + %d = %d\n", arg1, arg2, res);
}
```

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.n.h>
#include <sys/mman.h>

int main(int argc, char *argv[]) {
    void *memory = mmap(0, 4096, PROT_READ | PROT_WRITE | PROT_EXEC,
                        MAP_PRIVATE | MAP_ANONYMOUS, -1, 0);

    unsigned char code[] = {
        0x55, 0x48, 0x89, 0xe5, 0x89, 0x7d, 0xfc, 0x89, 0x75,
        0xf8, 0x8b, 0x45, 0xfc, 0x03, 0x45, 0xf8, 0x5d, 0xc3
    };
    memcpy(memory, code, sizeof(code));
    int (*func)(int, int) = memory;

    int arg1 = atoi(argv[1]);
    int arg2 = atoi(argv[2]);
    int res = func(arg1, arg2);
    printf("%d + %d = %d\n", arg1, arg2, res);
}
```

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.n.h>
#include <sys/mman.h>

int main(int argc, char *argv[]) {
    void *memory = mmap(0, 4096, PROT_READ | PROT_WRITE | PROT_EXEC,
                        MAP_PRIVATE | MAP_ANONYMOUS, -1, 0);

    unsigned char code[] = {
        0x55, 0x48, 0x89, 0xe5, 0x89, 0x7d, 0xfc, 0x89, 0x75,
        0xf8, 0x8b, 0x45, 0xfc, 0x03, 0x45, 0xf8, 0x5d, 0xc3
    };
    memcpy(memory, code, sizeof(code));
    int (*func)(int, int) = memory;

    int arg1 = atoi(argv[1]);
    int arg2 = atoi(argv[2]);
    int res = func(arg1, arg2);
    printf("%d + %d = %d\n", arg1, arg2, res);
}
```

```
$ ./userProgram 10 15

10 + 15 = 25
```

more to read:

LWN: How programs get run Janina Mincer-Daszkiewicz : Process address space

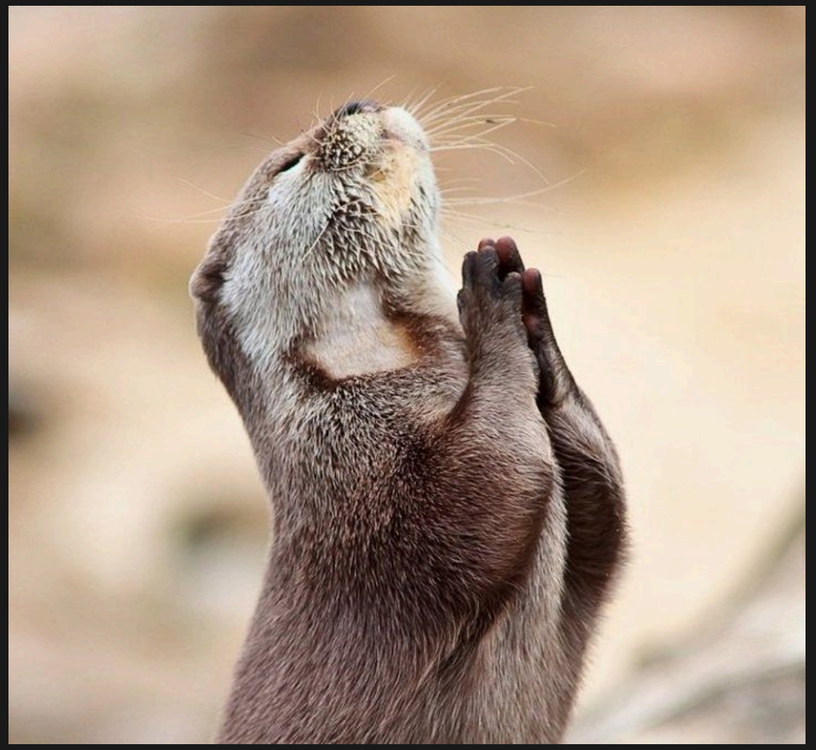
Adrian Huang: Process Address Space

more to watch:

https://youtu.be/dOfucXtyEsU https://youtu.be/OGPmZzhDPYw

https://youtu.be/fGnbGX88z3Y

Thank you for your attention!



Any questions?