Programando Assembly no Linux x64

+ O arquivo com nossas syscalls será o
/usr/include/x86_64-linux-gnu/asm/unistd_64.h
→ dessa vez a write será setada como 1
→ invés da interrupção 0x80, usaremos syscall
→ site para arquitetura x64: https://syscalls.33challs.com/?arch=x86_64

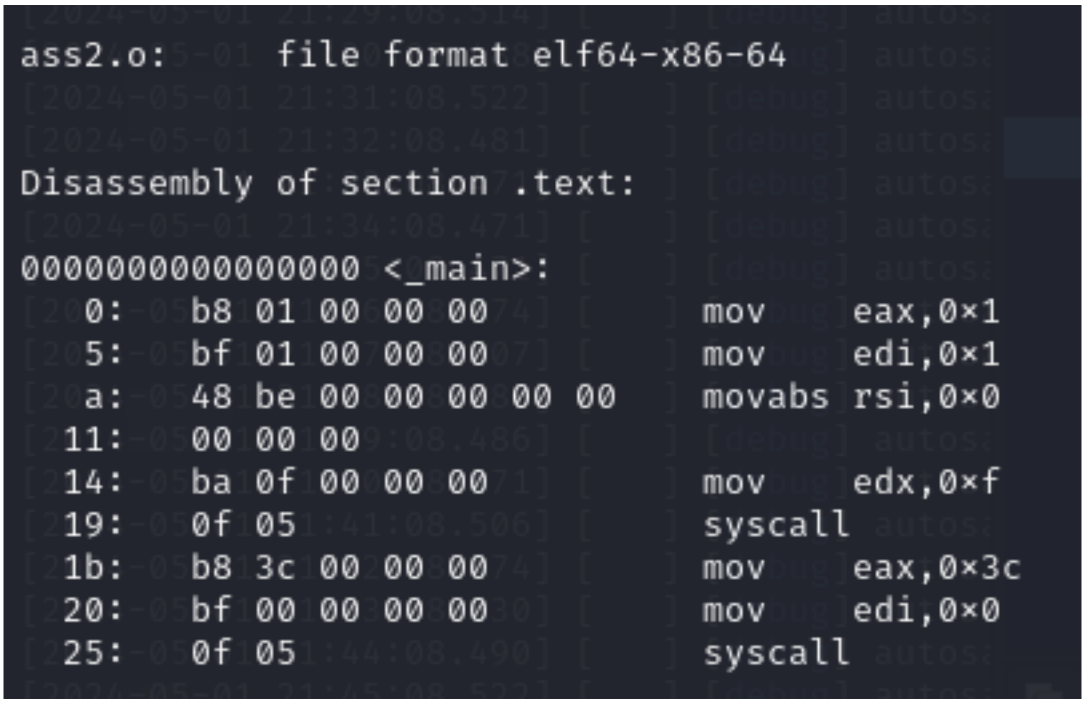
```
-------------------------------ass2.asm-----------------------------------------
global _main

section .data
	curso: db "Desec Security", 0xa

section .text

_main:
	mov rax, 1 ; a write em 64 bits é 1
	mov rdi, 1 ; nosso file descriptor que é 1 tbm
	mov rsi, curso
	mov rdx, 15 ; tamanho
	syscall

	mov rax, 60; a exit aq é 60
	mov edi, 0
	syscall
-------------------------------------------------------------------------------------
```

```
nasm -f elf64 ass2.asm
```

```
file ass2.o
```

ass2.o: ELF 64-bit LSB relocatable, x86-64, version 1 (SYSV), not stripped
→ Vemos que de fato ele gerou um 64-bit

```
objdump -d -M intel ass2.o
```

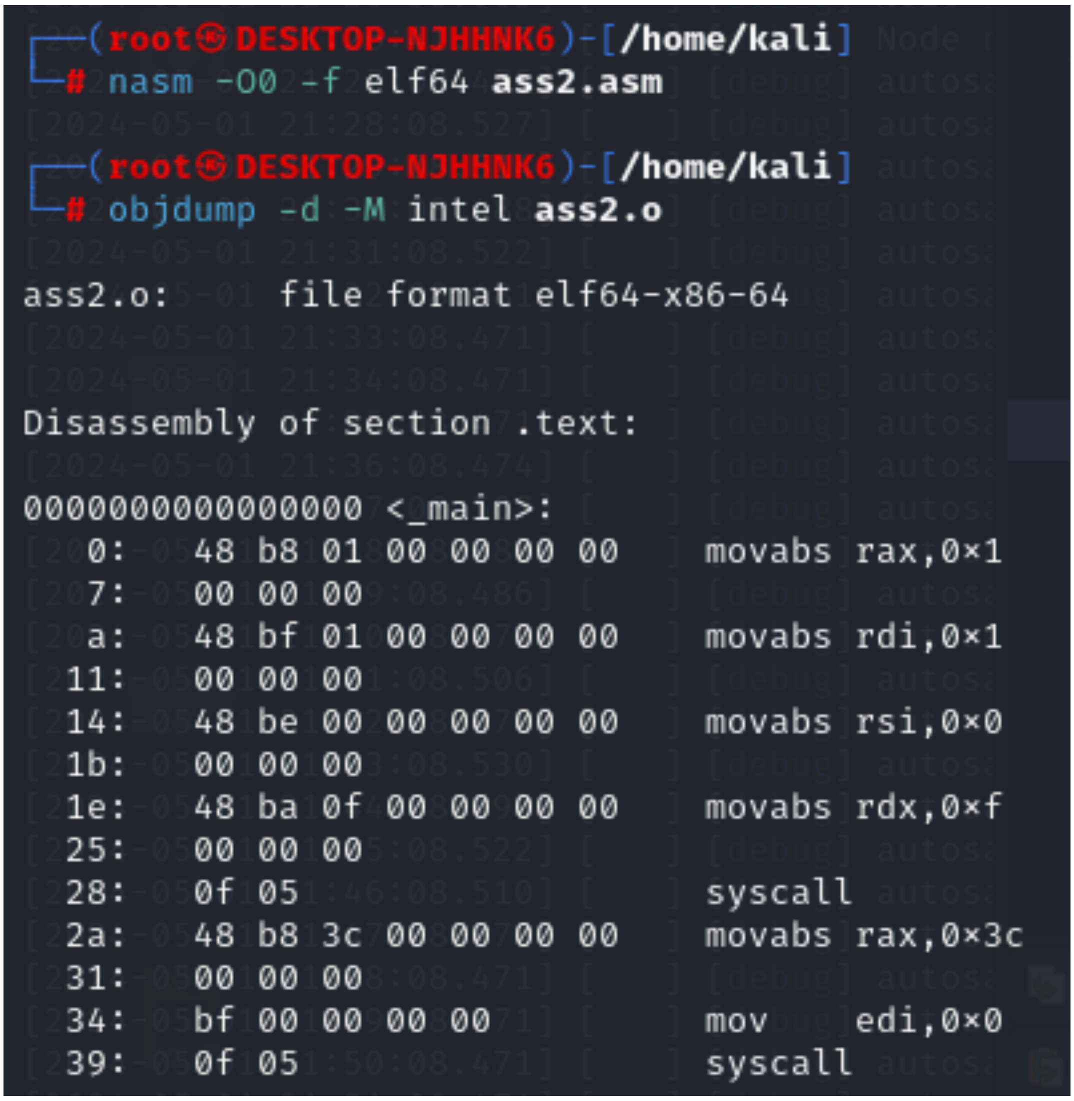


```
ass2.o:     file format elf64-x86-64

Disassembly of section .text:

0000000000000000 <_main>:
   0:   b8 01 00 00 00          mov    eax,0x1
   5:   bf 01 00 00 00          mov    edi,0x1
   a:   48 be 00 00 00 00 00    movabs rsi,0x0
  11:   00 00 00
  14:   ba 0f 00 00 00          mov    edx,0xf
  19:   0f 05                   syscall
  1b:   b8 3c 00 00 00          mov    eax,0x3c
  20:   bf 00 00 00 00          mov    edi,0x0
  25:   0f 05                   syscall
```

→ Por mais que tenhamos usado um formato 64-bit, ele otimizou para apenas 32-bit pelo motivo de termos feito uso de pouca informação

(tamanho pequeno)
→ Se quisermos que ele gere de fato um 64-bits:

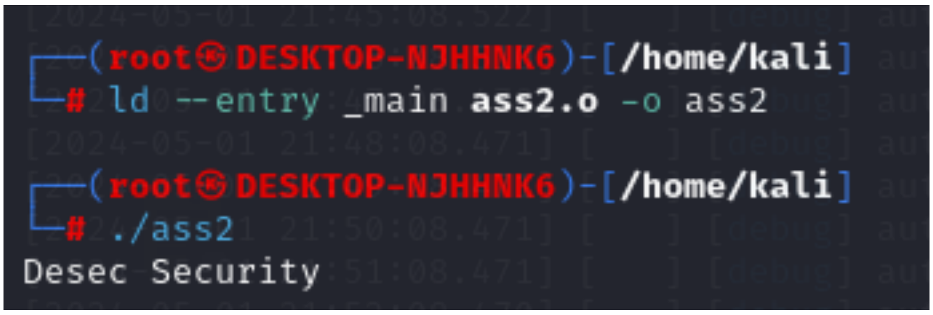
```
nasm -O0 -f elf64 ass2.asm
```



```
┌──(root㉿DESKTOP-NJHHNK6)-[/home/kali]
└─# nasm -O0 -f elf64 ass2.asm

┌──(root㉿DESKTOP-NJHHNK6)-[/home/kali]
└─# objdump -d -M intel ass2.o

ass2.o:     file format elf64-x86-64


Disassembly of section .text:

0000000000000000 <_main>:
   0:   48 b8 01 00 00 00 00    movabs rax,0x1
   7:   00 00 00
   a:   48 bf 01 00 00 00 00    movabs rdi,0x1
  11:   00 00 00
  14:   48 be 00 00 00 00 00    movabs rsi,0x0
  1b:   00 00 00
  1e:   48 ba 0f 00 00 00 00    movabs rdx,0xf
  25:   00 00 00
  28:   0f 05                   syscall
  2a:   48 b8 3c 00 00 00 00    movabs rax,0x3c
  31:   00 00 00
  34:   bf 00 00 00 00          mov    edi,0x0
  39:   0f 05                   syscall
```

→ Para linkar:

```
ld --entry _main ass2.o -o ass2
```



```
┌──(root㉿DESKTOP-NJHHNK6)-[/home/kali]
└─# ld --entry _main ass2.o -o ass2

┌──(root㉿DESKTOP-NJHHNK6)-[/home/kali]
└─# ./ass2
Desec Security
```