

Садржај излазне датотеке

Асемблерска линија	Адреса	Садржај							
enter 0,0	0	x	x	x	x				
mov rbx, la1	4	x	x	x	x	00	00	00	00
cmp rax, [rbp+0x12]	8	x	x	x	x				
call var[rip]	10	x	x	fc	ff	ff	ff		
jle la1	16	x	06						
mov rax, [rbp+0x12]	18	x	x	x	x				
push la2	1C	x	03						
mov rax, rbx	1E	x	x	x					
leave	21	x							
ret	22	?	?	?					
.quad la1	0	00	00	00	00	00	00	00	00
.quad la2-4	8	1d	00	00	00	00	00	00	00

text

```

.intel_syntax noprefix
.section .text
.global min, la1
min: enter 0, 0      # 0x00
    mov rbx, la1     # 0x04
    cmp rbx, [rbp]+0x12 # 0x0c
    call var[rip]    # 0x10
    jle la1          # 0x16
    [mov rax, [rbp]+0x12 # 0x18
    jmp la2          # 0x1c
la1: mov rax, rbx    # 0x1e
la2: leave          # 0x21
    ret              # 0x22

.data
.global var
.type var, @object
var:
.quad la1, la2-4
.end

```

1C
18
—
4

14

1E
18
—
06

21
1E
—
3

14

Напомена: хексадецимални бројеви наведени у коментару изворног асемблерског кода представљају померај до почетка инструкције у том реду гледано од почетка секције којој инструкција припада. Такође, усвојити претпоставку да је величина операционог кода:

- 2B за инструкцију `call` за PC релативно адресирање,
- 4B за инструкцију `mov` за апсолутно адресирање и
- 1B за инструкције `jle` и `jmp` за PC релативно адресирање.

A
B
C
D
E
F

Интерна табела симбола

Симбол	Секција	Вредност	Видљивост
min	.text	0	G
la1	.text	1E	G
la2	.text	21	L
var	.data	0	G
.text	.text	0	L

Табела обраћања унапред

Секција	Локација	Акција	Симбол

$$la2 = .text + 21$$

$$\begin{aligned}
 la2 - 4 &= (.text + 21) - 4 \\
 &= .text + (20 - 3) \\
 &= .text + 1d
 \end{aligned}$$

ЕЛФ табела симбола

[illegible]

ЕЛФ релокациони записи .text секције

[illegible]

ЕЛФ релокациони записи .data секције

[illegible]