


- 
1. Faça um código que receba uma string de entrada e imprima a sua primeira e última letra. Utilize o conceito de pilha para resolver esta questão.

CALL - RET



O **RSP** é o *stack pointer* e sua função é guardar o endereço do topo da pilha.

O **RBP** é o *base pointer* e sua função é guardar o endereço da base da pilha.

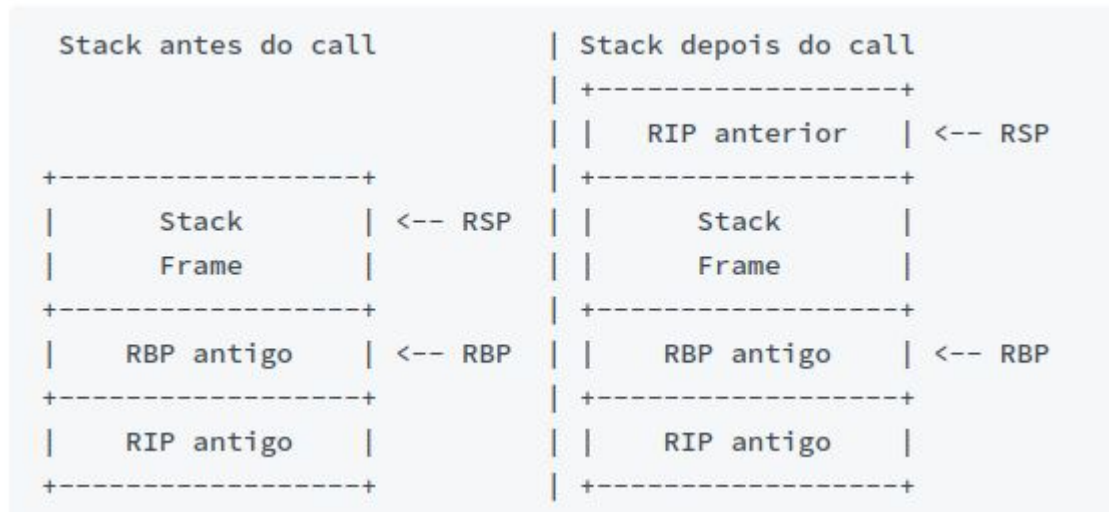
O **RIP** indica qual endereço será executado.

call

Empilha o RIP atual na pilha.

ret

Coloca o topo da pilha no ponteiro de instrução.



push rbp # Salva o rbp antigo

mov rbp, rsp # rsp é movido para rbp para criar um novo quadro de pilha para a função call

```
1 section .data
2
3 espaço db " "
4
5 section .bss
6
7 nome resb 10
8 last_letra resb 1
9
10
11 section .text
12
13 ▾ strlen:
14     mov rax, 0
15 ▾     .loop:
16         cmp byte [rdi+rax], 0
17         je .end
18
19         inc rax
20         jmp .loop
21 ▾     .end:
22         ret
23
```

```

25 global _start
26 _start:
27
28     mov rax, 0
29     mov rdi, 0
30     mov rsi, nome
31     mov rdx, 4
32     syscall
33
34     mov rdi, nome
35     call strlen
36     push rax ; Inserindo valor de rax na pilha
37
38     mov rax, 1
39     mov rdi, 1
40     mov rsi, nome
41     mov rdx, 1
42     syscall
43
44     pop rcx ; Colocado o valor que está no topo da pilha em rcx
45     mov rdi, [nome+rcx-1]
46     mov [last_letra], rdi
47

```

```
47
48     mov rax, 1
49     mov rdi, 1
50     mov rsi, espaco
51     mov rdx, 1
52     syscall
53
54     mov rax, 1
55     mov rdi, 1
56     mov rsi, last_letra
57     mov rdx, 1
58     syscall
59
60     mov rdi, 0
61     mov rax, 60
62     syscall
63
64
65
```

casa

Saída do programa

c a

[Execution complete with exit code 0]

