



Atividade 3 - Registradores

Até este ponto da disciplina, trabalhamos e estudamos os registradores de uso geral e os compreendemos os registradores de uso específico.

1º Liste os registradores de uso geral da arquitetura Intel 64 bits e a descrição de cada.

Resposta:

0	rax	acumulador
1	rcx	contador
2	rdx	data
3	rbx	base
4	rsp	stack pointer
5	rbp	base pointer
6	rsi	source index
7	rdi	destination index
8	r8	-
9	r9	-
10	r10	temporário
11	r11	temporário
12	r12	-
13	r13	-
14	r14	-
15	r15	-

2° Quais são os registradores especiais que têm 128 bits de tamanho? descreva a tecnologia SSE.

Resposta: xmm0 - xmm15. Originalmente a tecnologia SSE permitia que esses novos registradores fossem utilizados para armazenar 4 pontos flutuantes de 32 bit cada um e precisão única. Porém hoje em dia podem ser utilizados para diversos armazenamentos, como:

- 2 pontos flutuantes de 64 bit (e precisão dupla)
- 2 inteiros de 64 bit
- 4 pontos flutuantes de 32 bit (e precisão única)
- 4 inteiros de 32 bit
- 8 inteiros de 16 bit
- 16 caracteres de 8 bi (bytes)

3° Qual uso do registrador *rip* e qual sua função?

Resposta: É o registrador que aponta para a instrução seguinte a ser executada. Não pode ser acessado diretamente pela pessoa programadora mas é muito utilizado para fazer endereçamento variável de variáveis globais, onde não é necessário sempre carregar o programa e variáveis no mesmo endereço de memória, tornando-o assim mais difícil de ser atacado.

4° *rflags*, qual função desse registrador? O que significam as flags CF, AF, ZF, OF e SF?

Resposta: Indicam o estado do microprocessador após a execução de cada instrução e são subconjuntos de bits individuais, cada um indicando uma propriedade.

São eles:

Nome	Sigla	Função/Significado
Carry Flag	CF	Indica o “vai-um”
Auxiliary Carry	AF	Indica número par de 1’s no byte inferior
Zero Flag	ZF	Indica “zero” na última operação
Overflow Flag	OF	Indica erro de transbordamento
Sign Flag	SF	Indica resultado negativo

5° Qual registrador armazena o endereço que está no topo da pilha?

Resposta: o stack pointer/r4/rsp

6º Segundo código abaixo qual valor estará em `rsp`? por que isso ocorre?

```
1 section .data
2     msg db 5,6,7
3
4 section .text
5     global _start
6
7 _start:
8
9     push 5
10    push 6
11    push 7
12
13
14    mov rdi, [rsp]
15    mov rax, 60
16    syscall
17
18
```

Resposta: O valor 7. Pois ele foi o último número a ser empilhado - passado para a stack, através do comando `push 7`.

7º Reelabore o código acima inserindo o valor 5 no registrador `rax`.

```
1 section .data
2     msg db 5,6,7
3
4 section .text
5     global _start
6
7 _start:
8     push 5
9     push 6
10    push 7
11
12    mov rdi, [rsp]
13    mov rax, 5
14    syscall
```