

Bem-vindos ❤️

RECURSO



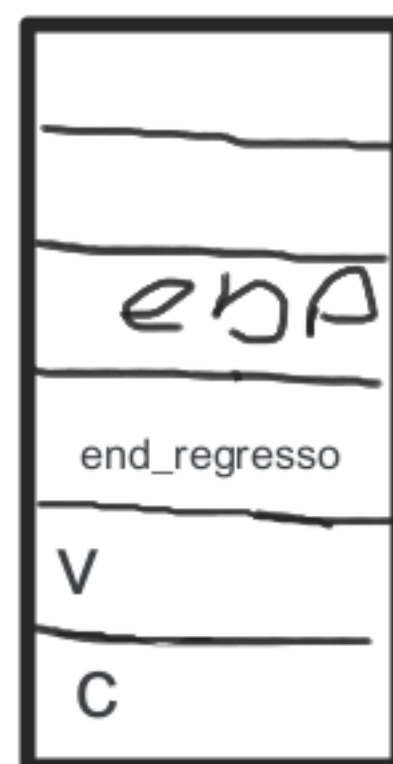


Complete o fragmento de código em C com valores, operações e variáveis, tendo em atenção o código de montagem gerado pelo compilador.

Nota: A primeira caixa é uma variável, na segunda uma operação (exemplo +) e na última uma constante.

```
int f( int v, int c){  
    return    ;  
}
```

```
f:  
    pushl    %ebp  
    movl     %esp, %ebp  
    movl     12(%ebp), %eax  
    andl     $33, %eax  
    leave  
    ret
```



Considere na arquitetura de um processador IA-16 a adição de 2 inteiros (em complemento para 2), `addw %ax, %bx`.

Considerando que o conteúdo desses registros antes da adição é `0x6cd` (em %ax) e `0xf8ad` (em %bx), **mostre** o conteúdo dos seguintes registros no final da execução desta soma:

• em %ax **em octal** (6 dígitos, não indique a base):

• o valor representado **em decimal** no registro %bx:

$$bx = bx + ax$$

$$ax = 0x6cd = 3315_8 = 3480_{10}$$

$$bx = 0xf8ad = 6366_{10}$$

$$\underline{\hspace{1cm}} +$$

$$67193_{10}$$

Handwritten binary representation of 0xf8ad (11011000101101010101010101010101) with red brackets and an arrow pointing to the decimal conversion of 0x6cd.

Considere a representação de números, em vírgula flutuante com 12 bits, baseada na norma IEEE. O expoente é representado por 4 bits, em excesso de  $2^{(n-1)}-1$ , enquanto os últimos 7 bits representam a mantissa.

Indique o valor (em decimal) representado pelo padrão binário "1110 1010 0111":

Mostre, usando o padrão binário baseado na norma IEEE, a representação do valor  $1/32$   $k_1$ :

$$\begin{array}{c} \text{S} \quad \text{E} \quad \text{M/F} \\ \text{---|---} \\ V = (-1)^S \times 1, \text{F} \times 2^{E-7} \end{array}$$

a)

'1110 1010 0111'

$$V = (-1)^{13_{10}} \times 1,0100111 \times 2^{13-7} = -83,5$$

b)

$$1/32 = 2^{-5} \times 2^{10} = 2^5 = 1,0 \times 2^5$$

01 1100 | 000 000 000

Exp - 7 = 5  
 $\therefore \text{Exp} = 12$   
 1100

Considere o estado parcial de uma máquina com um processador da família IA-32 (little endian) , ilustrado abaixo.

Registos	Memória (código)	Memória (dados)
%eip = 0x00004053	0x4050 pushl %ebp	0x04400: 0xAA 0x00 0x00 0x01
%ebp = 0x00004404	0x4051 movl %esp, %ebp	0x04404: 0x10 0x70 0x00 0x00
%eax = 0x00000100	0x4053 andl 12(%ebp), %eax	0x04408: 0xB5 0x40 0x00 0x00
%ebx = 0xFFFFFFFF	0x4056 movl \$4, %eax	0x0440C: 0x20 0xA4 0x11 0x00
%ecx = 0x00000001	0x4058 leave	0x04410: 0x01 0xA4 0xFF 0x00
%esp = 0x00004404	0x4059 ret	

Apresente, em hexadecimal ), o 1º bloco de informação que circula nos seguintes barramentos, apenas para a execução da instrução em 0x4053 (a instrução já se encontra decodificada):

- no barramento de endereços: 0x
- no barramento de dados: 0x

nao tenho a certeza



Considere o estado parcial de uma máquina com um processador da família IA-32 (little endian), ilustrado abaixo.

### Registos

%eip = 0x00004053  
%ebp = 0x0000FF04 ←  
%eax = 0x00000100  
%ebx = 0xFFFFFFFF  
%ecx = 0x00000001  
%esp = 0x0000FF04 — 4

### Memória (código)

0x4050 pushl %ebp  
0x4051 movl %esp, %ebp  
0x4053 movl \$0x0300, %eax  
0x4056 andl 12(%ebp), %eax  
0x4059 leave  
0x405A ret

### Memória (dados)

0x0FF00: 0xAA 0x00 0x00 0x01  
0x0FF04: 0x10 0x70 0x00 0x00  
0x0FF08: 0xB5 0x40 0x00 0x00  
0x0FF0C: 0x20 0x51 0x11 0x00  
0x0FF10: 0x01 0x3D 0xFF 0x00

Indique, os seguintes valores em hexadecimal (sem escrever o "0x" no início):

- o conteúdo do registo %eax no final da execução desta função: 0300
- o endereço da instrução que o processador vai executar após a instrução de "ret": 0000FF04

00FF3001, 0A0300