

Imprimir as folhas antes da entrega do ficheiro devido a gralhas, por isso, alterei os valores emados pelos comentários, segundo o ficheiro corrigido.

Nº 07368

Nome: João Pedro Vilas Boas Braga

Turma: 11E1 725

Resolução dos exercícios

1. (A) Acesso a operandos

| Operando | Valor | Comentário |
|----------------|-------|--------------|
| %eax | 0x200 | Registo %eax |
| 0x204 - 0x204 | 0xCB | Mem[0x204] |
| 0x208 - 0x208 | 0x208 | Constante |
| (%eax) | 0xDD | Mem[0x200] |
| 4(%eax) | 0xCB | Mem[0x204] |
| 9(%eax,%edx) | 0x10 | Mem[0x20c] |
| 0x1fc(,%ecx,4) | 0xDD | Mem[0x200] |
| (%eax,%edx,4) | 0x10 | Mem[0x20c] |

2. (R) Transferência de informação em funções

(Resduido nas outras páginas)

3. (R) Load effective address

| Instrução | Valor |
|---------------------------|---------------------|
| leal 6(%eax), %edx | $z = 6 + x$ |
| leal (%eax,%ecx), %edx | $z = x + y$ |
| leal (%eax,%ecx,8), %edx | $z = x + 8 * y$ |
| leal 7(%eax,%eax,4), %edx | $z = x + 4 * x + 7$ |
| leal 6(%eax,%ecx,4), %edx | $z = x + 4 * y + 6$ |

4. (A) Operações aritméticas

| Instrução | Destino | Valor |
|--------------------------|--------------|-------|
| subl %edx,4(%eax) | Mem[0x204] | 0xCB |
| imull \$16,(%eax,%edx,4) | Mem[0x20c] | 0x100 |
| incl 8(%eax) | Mem[0x208] | 0x15 |
| decl %ecx | Registo %ecx | 0x0 |

9. Controlo do fluxo de execução de instruções

- a) 8048d1c: 7d 9e f8 jge XXXXXXXX _____
- b) XXXXXXX: eb 54 jmp 8047c42 _____
- c) 8048902: e9 c2 10 00 00 jmp XXXXXXXX _____

① $(\%eax) \rightarrow$ Espaço da memória cujo endereço é o valor ~~em~~ guardado no registo $\%eax$
 \downarrow
 $0x200$

$$4(\%eax) \rightarrow 0x200 + 4 = 0x204$$

$$\begin{aligned} 8(\%eax, \%edx) &\rightarrow \text{Mem}[\text{Reg}(\%eax) + \text{Reg}(\%edx) + 8] \\ &= \text{Mem}[0x200 + 0x3 + 0x8] \\ &= \text{Mem}[0x20c] \end{aligned}$$

$$\begin{aligned} 0x1fc(, \%ecx, 4) &= \text{Mem}[4 * \text{Reg}(\%ecx) + 0x1fc] \\ &= \text{Mem}[4 * 0x1 + 0x1fc] \\ &= \text{Mem}[0x200] \end{aligned}$$

$$\begin{array}{r} 11 \\ 0x1fc \\ 0x004 \\ \hline 0x200 \end{array}$$

$$\begin{aligned} (\%eax, \%edx, 4) &= \text{Mem}[\text{Reg}(\%eax) + 4 * \text{Reg}(\%edx)] \\ &= \text{Mem}[0x200 + 4 * 0x3] \\ &= \text{Mem}[0x20c] \end{aligned}$$

② Void decode 1 (int *xp, int *yp, int *zp); {

```

int x = *xp;
int y = *yp;
int z = *zp;
*xp = x;
*yp = y;
*zp = z;
}

```

$$\begin{aligned} \&xp &= 8(\%ebp) \\ \&yp &= 12(\%ebp) \\ \&zp &= 16(\%ebp) \end{aligned}$$

③

| Registo | Variável |
|---------|----------|
| $\%eax$ | x |
| $\%ecx$ | y |
| $\%edx$ | z |

$\text{leal } (\%eax, \%ecx), \%edx \rightarrow z = x + y$
 $\text{leal } (\%eax, \%ecx, 8), \%edx \rightarrow z = x + 8 * y$
 $\text{leal } 7(\%eax, \%eax, 4), \%edx \rightarrow z = x + 4 * x + 7$
 $\text{leal } 6(\%eax, \%ecx, 4), \%edx \rightarrow z = x + 4 * y + 6$

④ `subl %edx, 4(%eax)`
 ↓
 subtract (→) long (→ 4 bytes) em 1A-32
 ↳ 0x3
 $0xC8 - 0x3 = 0xC8$

$$\begin{aligned} 4(\%eax) &= \text{Mem}[\text{Reg}[\%eax] + 4] \\ &= \text{Mem}[0x200 + 0x4] \\ &= \text{Mem}[0x204] \end{aligned}$$

`imull $16, (%eax, %edx, 4)`

↓
constante 16

$$\begin{aligned} &\rightarrow \text{Mem}[\text{Reg}(\%eax) + 4 * \text{Reg}(\%edx)] \\ &= \text{Mem}[0x200 + 4 * 0x3] \\ &= \text{Mem}[0x20c] \end{aligned}$$

$$0x10 * 16 = 0x100 \quad \rightarrow \text{valor } 0x10$$

`incl 8(%eax)`

$$\begin{aligned} &\rightarrow \text{Mem}[\text{Reg}(\%eax) + 8] \\ &= \text{Mem}[0x200 + 8] = \text{Mem}[0x208] \end{aligned}$$

$$\rightarrow \text{Valor } 0x14$$

$$\text{incl} \rightarrow 0x14 + 1 = 0x15$$

`decl %ecx`

$$\rightarrow \text{Reg}[\%ecx]$$

$$\rightarrow \text{valor } 0x1$$

$$\rightarrow 0x1 - 0x1 = 0x0$$

⑤ `Not sei`