

## Laborator 2

### • Operatii aritmetice si logice

$$\text{add } op_1, op_2 \equiv op_2 = op_2 + op_1$$

$$\text{sub } op_1, op_2 \equiv op_2 = op_2 - op_1$$

$$\text{mul } op \equiv (edx, eax) \equiv eax \cdot op$$

$$\text{imul } op \equiv (edx, eax) \equiv eax \cdot op$$

$$\text{div } op \equiv (edx, eax) = (edx, eax) / op$$

$$\text{idiv } op \equiv (edx, eax) = (edx, eax) / op$$

ex: x.leng 4  
y.leng 6  
z.leng 2<sup>31</sup>  
0x80000000

$$x \cdot y = 24$$

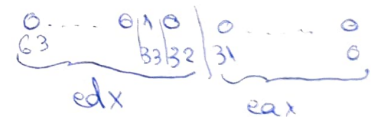
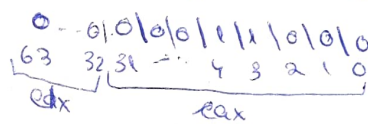
mov x, %eax

mul y →  $\begin{cases} eax = 24 \\ edx = 0 \end{cases}$

$$x \cdot z = 2^{33}$$

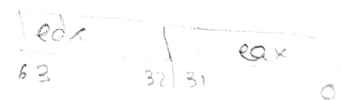
mov x, %eax

mul z →  $\begin{cases} eax = 0 \\ edx = 2 \end{cases}$



mov op, %eax  
mul op

$$(op + op) \cdot 2$$



$$op \cdot op = 2^{32} \cdot edx + eax$$

ex: div op

$$D = \frac{C}{R} + R$$

(edx, eax) ← eax ← edx  
                  ↓ op  
                  2<sup>32</sup> · edx + eax

12/4 → mov \$0, %edx  
          mov \$12, %eax  
          mov \$4, %edx  
          div %edx  
          eax = 1  
          edx = 5

### • Operatii logice

$$\text{not } op \equiv op = \neg op$$

$$\text{and } op_1, op_2 \equiv op_2 = op_2 \& op_1$$

$$\text{or } op_1, op_2 \equiv op_2 = op_2 | op_1$$

$$\text{xor } op_1, op_2 \equiv op_2 = op_2 \wedge op_1$$

## • Operații de shift / deplasare logică

$\text{shl nr, op} \equiv \text{op} = \text{op} \ll \text{nr}$   
 $\text{shr nr, op} \equiv \text{op} = \text{op} \gg \text{nr}$   
 $\text{sar nr, op} \equiv \text{op} = \text{op} \gg \text{nr}$   
 $\text{sal nr, op} \equiv \text{op} = \text{op} \ll \text{nr}$   $\rightarrow$  cu păstrarea semnului  
 $\text{r/l} = \text{left / right}$   
 $\text{a/b} = \text{op. pe biti sau aritmetice}$

## • Salt condiționat: ~~jmp et~~ jmp et

ex: main:

jmp et2

et2:

mov \$4, %eax

## • Salțuri condiționate (veri pef. asc. - lab.)

Obs: Pt a utiliza unul din op. aceia lbl. utilizată înainte inst.

cmp %eax, %ebx

jb et

$\downarrow$

dacă  $\%ebx < \%eax \rightarrow$  salt la et.

altfel cont. exec cu inst. urm. de după jb.

**loop et**

•  $\text{ecx} = \text{ecx} - 1$

•  $\text{if}(\text{ecx} \neq 0) \text{ goto et}$

ebx // cont - exec;

$4+3+2+1 \rightarrow$  cartul 2

$0+1+2+3+4 \rightarrow$  cartul 1