ASC

18.10.2023

Laborator 2

- Operații aritmetice și logice

add op1, op2 ≡ op2 = op2 + op1

sub op1, op2 ≡ op2 = op2 - op1

mul op ≡ (edx, eax) ≡ eax · op

imul op ≡ (edx, eax) ≡ eax · op

div op ≡ (edx, eax) = (edx, eax)/op

idiv op ≡ (edx, eax) = (edx, eax)/op

ex: x.long 4
y.long 6
z.long 2^{31}

0x800 0000

$x \cdot y = 24$

mov x, %eax

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

$\underbrace{\overset{0}{63} \dots \overset{0}{32}}_{edx}\ \underbrace{\overset{0}{31} \dots \overset{0}{\ } \overset{0}{\ } \overset{0}{\ } \overset{1}{4}\ \overset{1}{3}\ \overset{0}{2}\ \overset{0}{1}\ \overset{0}{0}}_{eax}$

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

mov x, %eax

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

$\underbrace{\overset{0}{63} \dots \overset{0}{\ }\ \overset{1}{33}\ \overset{0}{32}}_{edx}\ \underbrace{\overset{0}{31} \dots \overset{0}{0}}_{eax}$

mov op, %eax
mul op

$(\underbrace{op}_{< 2^{32}} + \underbrace{op2}_{< 2})$

edx	eax
63 32	31 0

$op \cdot op2 = 2^{32} \cdot edx + eax$

ex: div op

D = Î · C + R

(edx, eax) → $2^{32} \cdot edx + eax$; Î → op; C → eax; R → edx

12/7 → mov $0, %edx
mov $12, %eax
mov $7, %edx
div %edx
eax = 1
edx = 5

- Operații logice

not op ≡ op = ~op

and op1, op2 ≡ op2 = op2 & op1

or op1, op2 ≡ op2 = op2 | op1

xor op1, op2 ≡ op2 = op2 ^ op1

- Operații de shift / deplasare logică

shr număr, op ≡ op = op >> nr.

shl nr, op ≡ op = op << nr

sar nr, op ≡ op2 = op >> nr

sal nr, op ≡ op2 = op << nr |→ cu păstrare semn

r/l = left/right

a/h = op. pe biți sau aritmetice

- Salt necondiționat : ~~jmp et~~ jmp et

ex:

```
main :
        jmp et2
        . . . . . . . .
et2 :
      mov $4, %eax
      . . . . . . . .
```

- Salturi condiționate (vezi pdf asc.-lab.)

Obs: Pt a utiliza unul din op. aceea tb. utilizată înainte instr.:

```
cmp %eax, %ebx
jb et
```

↓

dacă %ebx < %eax → salt la et.

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

$\boxed{\text{loop et}}$

- ecx = ecx - 1
- if(ecx != 0) goto et

  else //cont_exec;

4+3+2+1 → cazul 2

0+1+2+3+4 → cazul 1