**Contorul de locaţii şi aritmetica de pointeri**

SEGMENT data

a db 1,2,3,4 ; 01 02 03 04

lg db $-a ; 04

db a-$ ; = -5 = FB

c equ a-$ ; = -6 = FA

db lg-a ; 04

db a-lg ; = -4 = FC

db [$-a] ; expression syntax error

db [lg-a] ; expression syntax error

lg1 EQU lg1-a ; lg1 = 0 – DE CE ?????

g34 dw c-2 ; -8

b dd a-start ; expression is not simple or relocatable !!!

dd start-a ; OK !!!!!!!! - va fi POINTER !!!!!

dd start-start1 ; OK !!! – pt ca e vorba despre 2 etichete din cadrul aceluiasi segment! – TD va fi SCALAR !!!!

**segment code use32**

start:

mov ah, lg1 ; AH = 0

mov bh, c ; BH = -6 = FA

mov ch, lg ; OBJ format can handle only 16 or 32 byte relocation

mov ch, lg-a ; CH = 04

mov ch, [lg-a] ; mov byte ptr DS:[4] – Access violation – cel mai probabil…

mov cx, lg-a ; CX = 4

mov cx, [lg-a] ; mov WORD ptr DS:[4] – Access violation – cel mai probabil…

mov cx, $-a ; invalid operand type !!!!!

mov cx, a-$ ; OK !!!!!!

mov ch, $-a ; invalid operand type !!!!!

mov ch, a-$ ; OBJ format can handle only 16 or 32 byte relocation

mov cx, $-start ; ok !!!

mov cx, start-$ ; ok !!!

mov ch, $-start ; ok !!!

mov ch, start-$ ; ok !!!

mov cx, a-start ; ok !!!

mov cx, start-a ; invalid operand type !!!

start1:

mov ah, a+b ; MERGE !!!!!!!!!! DAR …. NU ESTE ADUNARE DE POINTERI !!!

a+b = (a-$$) + (b-$$)

Mov ax, (a-$$) + (b-$$)

mov ax, b+a ; AX = (b-$$) + (a-$$) – adunare de SCALARI !!!!

mov ax, [a+b] ; INVALID EFFECTIVE ADDRESS !!!! – ASTA CHIAR E ADUNARE DE POINTERI !!!!!!

**Concluzie:**

Expresiile de tip et1 – et2 (unde et1 si et2 sunt etichete – fie de cod, fie de date) sunt acceptate sintactic de catre NASM,

* Fie daca ambele sunt definite in acelasi segment
* Fie daca et1 apartine unui segment diferit fata de cel in care apare expresia, iar et2 este definita in segmentul in care apare expresia. Intr-un astfel de caz, TD asociat expresiei et1-et2 este POINTER si NU SCALAR (constanta numerica) ca si in cazul etichetelor ce fac parte din acelasi segment.

Scadere de offset-uri ce se raporteaza la acelasi segment = SCALAR

Scadere de pointeri din segmente diferite = POINTER