**Operands data type**

v d? ….

a d?...

b d?...

Push v - ok, stack 🡨offset v (pe 32 biti)

Mov eax, [v] ; EAX 🡨4 bytes from v

Push [v] - syntax error – Operation size not specified !

Push dword [v] – ok

Push [eax] – syntax error….

Push byte [eax] – syntax error….

Push word [eax] – ok

Push 15 ; PUSH DWORD 15

Pop [v] ; Op size not specified

Pop word [v]

Pop v ; sintaxa este POP destinatie; destinatie must be a L-value !!!!!

…dar v este R-value !!!! ; acest pop v este similar ca operatie cu 2:=3 !!!

Pop dword b ; syntax error !

Pop [eax] ; Op size not specified

Pop (d)word [eax] ; ok!

Pop 15 ; 15 is NOT a L-value !! – syntax error

Pop [15] ; syntax error - Op size not specified

Pop dword [15] ; syntactic ok , cel mai probabil run-time error deoarece probabil [DS:15] va provoca Access violation !!

Mov [v],0 ; syntax error - Op size not specified

Mov byte [v], 0 ; ok !

Mov [v], byte 0

Div [v] ; op. size ?

Div word [v]; ok !

Imul [v+2] ; op. size ?

Imul word [v+2]; DX:AX = AX\*word de la adresa v+2

Mov a,b ; syntax error pt ca a NU este L-value, ci R-value fiind un offset determinabil ca si constanta la momentul asamblarii

Mov [a], b ; syntax error – op. size not specified !

Mov byte [a], b ; syntax error – OBJ file 16 sau 32….

Mov word [a], b ; ok !! – 2 octeti inferiori din valoarea offset-ului lui b !

Mov dword [a], b ; full offset 32 bits

Mov a,[b] ; Invalid comb. Of opcode and operands (a = R-value)

Mov [a], [b] ; Invalid comb. Of opcode and operands (NU putem avea 2 operanzi simultan din memorie)

Mul v – syntax error – MUL reg/mem

Mul word v - syntax error – MUL reg/mem

Mul [v] - op. size not specified

Mul dword [v]; ok !

Mul eax ; ok !

Mul [eax] ; op. size not specified

Mul byte [eax] ; ok !!!

Mul 15 ; Invalid comb. Of opcode and operands - – MUL reg/mem

**Data definition directives**

OllyDbg – Intotdeauna DATA segment incepe la offset-ul 00401000

Segment data

a1 db 0,1,2,’xyz’ ; 00 01 02 ‘x’ ‘y’ ‘z’ – codurile asci (

78 79 7A – hexa !!!!!

db 300, “F”+3 ; 2c ‘cod ascii F + 3’

a2 TIMES 3 db 44h ; 44 44 44

a3 TIMES 11 db 5,1,3 ; 05 01 03… de 11 ori

a4 dw a2+1, ‘bc’ ; offset a2=00401008; 09 10 62 63

‘bc’ dif de ‘b’,’c’ ‘b’ ‘c’

a5 dd a2+1, ‘bcd’ ; 09 10 40 00 62 63 64 00

a6 TIMES 4 db ‘13’ ; 31 33 31 33 31 33 31 33

a7 db a2 ; syntax error !!!!! OBJ format can only handle 16- or 32- relocation (echiv. cu mov ah,a2)

a8 dw a2 ; 08 10

a9 dd a2 ; 08 10 40 00

a10 dq a2 ; 08 10 40 00 00 00 00 00

a11 db [a2] ; expression syntax error – pt ca [a2] NU este o expresie valida acceptata de catre asamblor, nereprezentand “o valoare constanta determinabila la momentul asamblarii” ! Dereferentierea implicate aici ested cee ace deranjeaza asamblorul !!

a12 dw [a2] ; expression syntax error

a13 dd dword [a2] ; expression syntax error

a14 dq [a2] ; expression syntax error

mov ax, v ; Warning – 32 bit offset in 16 bit field !!!

Specificatorii BYTE / WORD / DWORD / QWORD au intotdeauna doar rol de a clarifica o ambiguitate (inclusiv cand este vorba despre o variabila de memorie, faptul de a preciza mov BYTE [v], 0 sau mov WORD [v], 0 este tot o clarificare a ambiguitatii, cum nasm nu asociaza faptul ca v este byte/ word / dword).

mov [v],0 ; syntax error – operation size not specified

**Specificatorul QWORD nu intervine niciodata explicit** in cod pe 32 de biti.

Exemple unde e necesar un specificator de dimensiune al operanzilor:

    - mov [mem], 12

- (i)div [mem] ; (i)mul [mem]

* push [mem] ; pop [mem]
* push 15 - aici este o inconsistenta in NASM, asamblorul nu va emite eroare/warning ci va face push DWORD 15   
    
  Exemple de operanzi IMPLICITI efectiv pe 64 biti (in cod pe 32):
* mul dword [v] ; inmulteste eax cu dword-ul de la adresa v si depune in EDX:EAX rezultatul

    - div dword [v] ; impartire EDX:EAX la dword [v]

Discutii consultatii

File handle (=file ID) = file descriptor

PUSHF

POP EAX

…………

PUSH EAX

POPF

Mov ah, 10011101 ; pune in AH ultimul octet din reprezentarea numarului zecimal 10011101

.RADIX 2 (specific TASM).

Bits 16

…nasm….

‘a’ “a”

Sizeof(‘a’) = 1 96

Sizeof(“a”) = 2 96 00

ASCIIZ