

Lab 12: Observații:

!!! atentie, ca sir [ESI] nu funcționează în OLLY !!!!, doar scris sub forma: mov AL, [sir+ESI]
In NASM nu exista directiva offset !!! Aici e foarte important daca punem sau nu punem [] !!!

Dacă se folosește **mov EAX, dword sir** ; atunci va depune în EAX adresa unde începe sir în memorie

Dimensiunea registrului, CPU va cunoaște dimensiunea la care vrem să accesăm variabila din memorie

Dacă se folosește **mov EAX, [sir]** ; atunci va depune în EAX primul element al șirului, de tip doubleword ;

! Nu este obligatoriu, în acest caz, să scriem mov EAX, dword [sir]

ca să accesăm următorul element al șirului, va trebui să îl accesăm ca [sir +4]

Dacă se folosește **mov AX, [sir]** ; atunci va depune în AX primul element al șirului, de tip word

! Nu este obligatoriu, în acest caz, să scriem mov AX, word [sir]

ca să accesăm următorul element al șirului, va trebui să îl accesăm ca [sir +2]

Dacă se folosește **mov AL, [sir]** ; atunci va depune în AL primul element al șirului, de tip byte

! Nu este obligatoriu, în acest caz, să scriem mov AL, byte [sir]

ca să accesăm următorul element al șirului, va trebui să îl accesăm ca [sir +1]

Specificatorii folosiți de către funcțiile printf și scanf sunt:

Specificator	Ce se afișează	Exemplu	Dimensiune reprezentare valoare
c	Caracter	a	byte
d sau i	Întreg zecimal cu semn	392	dword
u	Întreg zecimal fără semn	7235	dword
x	Număr în hexazecimal fără semn	7fa	dword
s	String (șir de caractere, terminat cu 0)	exemplu	șir de bytes terminat în 0

Exemple de programe:

Exemplul 1: Citeste un caracter de la tastatura si il afiseaza, folosind specificatorul %c:

```
bits 32
global start

extern exit, scanf, printf
import exit msvcrt.dll
import scanf msvcrt.dll
import printf msvcrt.dll

segment data use32 class=data
a dd 0
formatCit db '%c',0
formatScr db 'caracterul este: %c',0
mesaj db 'Introduceti un caracter:',0
segment code use32 class=code
start:
; afiseaza mesaj introductiv
push dword mesaj
call [printf]
add esp, 4*1
; citeste un caracter de la tastatura
push dword a
push dword formatCit
call [scanf]
add esp, 4*2
; afiseaza caracterul citit
push dword [a]
push dword formatScr
call [printf]
add esp, 4*2

push dword 0
call [exit]
```

Exemplul 2: Citeste un string (șir de caractere) de la tastatura si il afiseaza, folosind specificatorul %s:

! la întâlnirea caracterului spațiu

```
bits 32
global start

extern exit, scanf, printf
import exit msvcrt.dll
import scanf msvcrt.dll
import printf msvcrt.dll

segment data use32 class=data
sir times 20 db 0
formatCit db '%s',0
formatScr db 'caracterele sunt: %s',0
mesaj db 'Introduceti un sir de caractere:',0
segment code use32 class=code
start:
; afiseaza mesaj introductiv
push dword mesaj
call [printf]
add esp, 4*1
; citeste un caracter de la tastatura
push dword sir
push dword formatCit
call [scanf]
add esp, 4*2
; afiseaza caracterul citit
push dword sir
push dword formatScr
call [printf]
add esp, 4*2

push dword 0
call [exit]
```

Citeste de la tastatura 2 caractere folosind functia scanf :

bits 32

global start

extern exit, scanf, printf

import exit msvcrt.dll

import scanf msvcrt.dll

import printf msvcrt.dll

segment data use32 class=data

a dd 0

b dd 0

formatCit db ' %c',0

formatScr db 10,13,'caracterul este: %c',0

mesaj db 10,13,'Introduceti un caracter:',0

segment code use32 class=code

start:

; afiseaza mesaj introductiv

push dword mesaj

call [printf]

add esp, 4*1

; citeste un caracter de la tastatura

push dword a

push dword formatCit

call [scanf]

add esp, 4*2

; afiseaza caracterul citit

push dword [a]

push dword formatScr

call [printf]

add esp, 4*2

; afiseaza mesaj introductiv

push dword mesaj

call [printf]

add esp, 4*1

; citeste un caracter de la tastatura

push dword b

push dword formatCit

call [scanf]

add esp, 4*2

; afiseaza caracterul citit

push dword [b]

push dword formatScr

call [printf]

add esp, 4*2

push dword 0

call [exit]

