



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale  
2007-2013



# Platformă de e-learning și curriculum e-content pentru învățământul superior tehnic

## Programare în limbaj de asamblare

### **19. Declararea datelor în LA.**

**Declarare etichete în limbajul de asamblare.**

## Declararea datelor în limbaj de asamblare

Declararea datelor se realizează cu ajutorul unor pseudoinstrucțiuni care asigură: alocarea de memorie pentru date, specificarea tipului datelor și inițializarea datelor. Datele pot fi specificate prin nume, expresii sau șiruri de caractere, care se evaluează în procesul de asamblare la constante. Operația de alocare și inițializare a datelor are loc în procesul de asamblare; datele sunt înscrise în fișierul obiect, de unde vor fi preluate la execuție. Sintaxa este următoarea:

```
[ nume_variabila ]          tip          [ lista_expresii ] [;comentarii]
sau
[ nume_variabila ]          tip          [numar] DUP ([ lista_expresii ]
unde
```

- *nume\_variabilă* = identificator, construit după regulile specificate anterior;
- *tip* = mnemonica pseudoinstrucțiunii, care specifică implicit tipul datelor:
  - db – define byte; definește date de tip octet întregi, cu sau fără semn;
  - dw – define word; cuvânt de 16 biți ce definește date întregi, cu sau fără semn;
  - dd – define double; dublu cuvânt, de 32 de biți ce poate defini un pointer sau o valoare întreagă sau reală, cu sau fără semn;
  - dq – define quadruple; cuvânt de 64 de biți ce poate defini o valoare reală sau întreagă;
  - df – define float; real pe 6 octeți;
  - dp – define pointer; referință specifică pentru 386, reprezentată pe 6 octeți (segmentul pe 2 octeți, iar deplasamentul pe 4 octeți);
  - dt – define ten bytes; dată de 10 octeți de tip real sau zecimal împachetat;
  - *nume\_structură* – numele unei structuri utilizată pentru definirea unor date cu această structură;
- *listă\_expresii* = expresii, separate prin virgule, evaluate la constante, care reprezintă valorile folosite la inițializarea datelor definite.

Simbolul ? arată că datele respective nu se inițializează. Construcția este recursivă. Evaluarea expresiilor se face la asamblare.

- *număr* = expresie ce reprezintă factorul de multiplicare al listei de expresii care urmează după operatorul DUP.

Exemple:

```
a      db      -100; se pot defini valori în intervalul [-128, +127] sau [0, 255]
lit    db      „abcde“; se vor memora codurile ASCII ale caracterelor din listă:
        61h, 62h, 63h, 64h, 65h
aw     dw      -1000 ; domeniul de valori: [-32768, 32767], sau [0, 65535]
AB     dw      „AB“ ; va depune la adresa AB: 42h, 41h
adrAB  dw      AB ; se va depune deplasamentul etichetei AB, stabilit la asamblare
off_AB dw      offset AB ; deplasamentul etichetei AB, stabilit la locatare
seg_AB dw      seg AB; adresa segment de la locatare
rez    dw      ? ; conținut nedefinit
v1     dd      1.0 ; valoare reală definită pe 32 biți
v2     dd      1A2B3C4Dh ; valoare întreagă pe 32 biți
```

adr\_v1 dd v1 ; pointer pentru referirea lui v1

Pentru a multiplica definierea unor date se utilizează operatorul DUP, alături de db, dw etc. Sintaxa este următoarea:

< numar > DUP (expresie pt. valoarea inițială)

În acest fel se definește o structură formată dintr-un <număr> de elemente, iar un element poate fi expresie numerică, semnul ? dacă nu se specifică valoarea inițială, o listă de elemente sau alți operatori DUP. Exemple:

db	100 dup (0) ; multiplicarea valorilor initiale
db	2 dup (0, 3 dup (1)), 10, 25, 100)
dw	100 dup (5 dup (4), 7)
db	10 dup ('mesaj ', 0dh, 0ah)

### Declararea etichetelor

Etichetele identifică locații de memorie ale instrucțiunilor, operanzilor sau apeluri de proceduri. Ele au trei atribute: segment, deplasament (offset) și distanță (near sau far) pentru etichete de instrucțiune și nume de proceduri, sau tip (byte, word etc.) pentru etichete de operanzi.

Etichetele de tip near pot fi definite astfel:

```
etich_near: mov ax, bx
proc_alfa label near
calcul proc near
```

În acest ultim caz, dacă nu se specifică tipul, în mod implicit acesta va fi tot de tip near.

Exemple pentru tipul far:

```
beta proc far
etich_far label far
```

Această directivă nu incrementează contorul de program (\$).

În cazul unei etichete de operand se va specifica tipul acesteia, ca de exemplu în cazul declarării vârfului stivei din segmentul de stivă:

```
stiva segment
dw 100 dup (?)
varf_stiva label word
stiva ends
```

pentru a permite, ulterior, inițializarea registrului SP cu vârful stivei.