



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

37. Definirea de structuri și înregistrări (directivele *struc* și *record*).

Definirea unei structuri. Directiva **STRUC**

O structură este o descriere care asociază un nume și attribute (lungime, tip, etc.) la o colecție de câmpuri, de același tip sau de tipuri diferite (echivalent cu RECORD din Pascal sau STRUCT din C).

```
nume_structura    STRUC
                  <declarații și definiți de date>
nume_structura    ENDS
```

```
[nume_variabila] nume_structura <[expresie][,exp,....]>
[nume_variabila] nume_structura expr dup (<[expr][,expr,...]>)
```

expresie - evaluată la o constantă, valoarea inițială a câmpului;

expr. - reprezintă numărul de duplicate ale structurii.

Exemple de inițializare:

<> - asociază câmpurilor valorile definite inițial, în decl. STRUC;

<5> - primul câmp al struc. va fi 5, iar celelalte, nedefinite, se vor asocia valorile definite inițial;

<5,15> - se vor inițializa primele două câmpuri, cu 5 și respectiv 15;

<.,7> - cel de-al treilea câmp va lua valoarea 7, iar restul valorile definite inițial;

<1.,7> - primul și cel de-al treilea vor primi valorile 1 și respectiv 7, iar celelalte vor primi valorile definite inițial, la declararea structurii.

Dacă inițial, la definire, nu li s-au asociat valori, câmpurile respective vor lua valoarea 0.

Pentru accesul la un câmp al unei structuri se folosește sintaxa:

```
nume_variabila.nume_câmp sau [reg_baza].nume_câmp
```

```
Tip_persoana struc
```

```
    nume db    20    dup(' '); se inițializează cu blankuri
    prenume db  20    dup(' ')
    varsta  db   0
    ocupatie db  15    dup(' ')
```

```
Tip_persoana ends
```

```
p1 Tip_persoana <'Ionescu','Ilie',63,'inginer'>
```

```
p2 Tip_persoana <> ; copie a valorilor inițiale
```

```
p3 Tip_persoana <.,21,'student'> ; inițializare vârstă și ocupație
```

```
grupa Tip_persoana 25 dup(<.,20,'student'>)
```

Definirea unei înregistrări. Directiva **RECORD**

Înregistrarea permite definirea de biți individuali sau grupuri de biți, dintr-un octet sau cuvânt. Fiecare bit sau grup de biți este denumit câmp. La aceste câmpuri ne putem referi prin numele simbolice asociate.

```
nume_înregistrare RECORD nume_câmp : expr [=expr'][,..]
```

expr = numărul de biți alocați câmpului respectiv;

expr' = această clauză este opțională și reprezintă valoarea inițială asociată câmpului respectiv; dacă nu se specifică, va fi implicit, egală cu 0.

Suma valorilor expresiilor pentru numărul de biți (*expr*) trebuie să fie o constantă întregă, și dacă este <= 8, rezultă că s-a definit o înregistrare de tip octet, dacă este > 8 și <=16, rezultă că s-a definit o înregistrare de tip cuvânt, etc. Dacă sunt definiți mai puțini biți decât

dimensiunea unui octet (cuvânt, etc.), atunci câmpurile sunt aliniate la dreapta în cadrul tipului respectiv.

```
model_rec record X:3=7, Y:4=8, Z:9=257
```

pentru această declarație se va genera un cuvânt: 111.1000.100000001.

declarația acestui tip de date poate fi folosită pentru alocare:

```
inregistrari model_rec 100 dup (<,15,3>)
```

În urma acestei declarații se alocă, începând de la adresa *inregistrari*, 100 de cuvinte, ale căror câmpuri (X, Y, Z) vor avea valorile:

X = 111, păstrează valoarea de la definiție,

Y = 1111, inițializată la alocare,

Z = 000000011, inițializată la alocare.

Pentru astfel de date sunt definiți doi operatori:

MASK nume_câmp - furnizează o mască de biți pentru câmpul respectiv (biții câmpului respectiv sunt 1, iar toți ceilalți 0);

WIDTH nume_câmp - furnizează o constantă întregă egală cu numărul de biți ai câmpului respectiv, sau se poate specifica numele înregistrării și va furniza numărul de biți al acesteia.

Pentru o referire directă la numele câmpului, într-o instrucțiune, asamblorul va genera o valoare întregă ce reprezintă poziția în cadrul octetului (sau cuvântului, etc.). Cu alte cuvinte, reprezintă numărul de deplasări la dreapta, necesare pentru a aduce câmpul respectiv în biții cei mai puțin semnificativi ai înregistrării respective.

Exemple de utilizare:

```
1) mov dx, inregistrari [40] ; (DX) = al 21-lea cuvânt
   and dx, mask Y ; (DX) = 000.11111.0000000000
   mov cl, Y ; (CL) = 9
   shr dx, cl ; se deplasează câmpul Y la dreapta
   mov ax, model <1,7,127> ; (AX) = 001.00111.001111111.
   dim_X equ width X ; dim_X = 3
   dim_model equ width model ; dim_model = 16
```

2) Selectarea, dintre angajații unei firme, a bărbaților cu o vechime mai mare de cinci ani, necăsătoriți:

```
date_angajat record vechime:6, sex:1, stare:1
```

```
tip_angajat struc
```

```
    nume db 20 dup(' ')
```

```
    prenume db 20 dup(' ')
```

```
    date date_angajat < >
```

```
tip_angajat ends
```

```
nr_ang equ 100
```

```
angajat tip_angajat nr_ang dup (< >)
```

```
; .....
```

```
mov cx, nr_ang
```

```
lea bx, angajat
```

```
iar:
```

```
mov al, [bx].date
```

```
test al, mask sex ; 0 = femeie, 1 = bărbat
```

```
jz respins
```

```

test al, mask stare    ; 0 = căsătorit, 1 = necăsătorit
jz respins
mov cl, vechime
shr al, cl             ; (AL) =0,0,vechime(6 biți)
cmp al, 5             ; testare vechime mai mare de 5 ani
jb     respins;
                    ; cel selectat este referit de (BX)
                    ; .... afișare: nume & prenume
respins:
add bx, size tip_angajat    ; se trece la următorul angajat
loop iar     ....

```