UNIUNEA EUROPEANĂ
GUVERNUL ROMÂNIEI
Instrumente Structurale
2007-2013

Platformă de e-learning și curriculă e-content pentru învățământul superior tehnic

Programare în limbaj de asamblare

36. Structuri de date în LA.

Structuri de date în limbaj de asamblare

Definirea unei structuri. Directiva *STRUC*

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

nume_structura STRUC
<declarații şi definiri 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 şi respectiv 15;
<,,7> - cel de-al treilea câmp va lua valoarea 7, iar restul valorile definite inițial;
<1,,7> - primul şi cel de-al treilea vor primi valorile 1 şi 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 foloseşte sintaxa:

nume_variabila.nume_câmp sau ***[reg_baza].nume_câmp***

```
Tip_persoana  struc
        nume    db      20      dup(' ') ; se inițializează cu blancuri
        prenume         db      20      dup(' ')
        varsta          db      0
        ocupatie        db      15      dup(' ')
Tip_persana   ends
p1 Tip_persoana         <'Ionescu','Ilie',63,'inginer'>
p2 Tip-persoana         <>      ; copie a valorilor inițiale
p3 Tip_persoana <,,21,'student'> ; inițializare vârstă şi ocupație
grupa  Tip_persoana  25 dup(<,,20,'student'>)
```

Programul următor construieşte o listă de tip coadă, formată din caractere introduse de la tastatură, care va fi apoi tipărită.

```
.model small
.stack  100h
.data
        tip_coada       struc
                next    dw      0 ; lista este în cadrul aceluiaşi segment
                car     db      0 ; sau info db 10 dup(0), pt. alte info
        tip_coada       ends
        lung_coada equ 80 ; dimensiunea maximă a cozii
        coada  tip_coada        lung_coada    dup(<>)
```

```
.code
start:
        mov ax, @data
        mov ds, ax              ; inițializare registru segment
        lea bx, coada  ; inițializare adresa de început a cozii
        mov si, size    tip_coada       ; (SI) = număr octeți / structură
        mov cx, lung_coada  ; contor număr maxim de elemente
iar:
        mov ah, 1               ; citirea unui caracter
        int 21h
        cmp al, 0dh    ; test sfârşit citire
        jz tiparire             ; tipărirea cozii
        mov [bx].car, al        ; se depune caracterul în coadă
        mov ax, bx
        add ax, si              ; actualizarea adresei următorului element
        mov [bx].next, ax       ; legătura la următorul element al cozii
        mov bx, ax              ; (BX) = adresa următorului element
        loop iar                ; dacă nu s-a atins limita max. (80), se reia
        mov dl, 0dh    ; se revine la începutul liniei
        mov ah, 2               ; se transmite caracterul la display
        int 21h         ; pentru a tipări de la începutul liniei
tiparire:
        mov dl, 0ah    ; se trece pe o linie nouă
        mov ah, 2               ; funcția de tipărire a caracterului din (DL)
        int 21h
        lea bx, coada  ; inițializarea cu adresa de început a cozii
        mov cx, lung_coada ; număr maxim de caractere de tipărit
reia:
        cmp [bx].next, 0        ; test pt. ultimul element al cozii
        jz gata_tip             ; dacă câmpul next = 0, e ultimul element
        mov dl, [bx].car        ; tipărire caracter
        mov ah, 2
        int 21h
        mov bx, [bx].next       ; adresa următorului element din coadă
        loop reia
gata_tip:
        mov ax, 4c00h
        int 21h
end     start
```