

Struktúrák és makrók

Struktúrák

A struktúrák az assembly programozási nyelvben olyan összetett típusok, amelyek több mezőt tartalmazhatnak. A mezők definíciója csak egyszerű adatdefiníciós utasításokkal történhet. A struktúrák bizonyos mezői felülírhatók mások pedig nem. A struktúra mezeje abban az esetben felülírható, ha egyetlen adatot tartalmaz (pl. vesszővel elválasztott elemek nem írhatók felül).

A struktúrákat az alábbi séma szerint definiálhatjuk:

S	STRUC		; struktura típus definicio
F1	db 5		; felulirhato
F2	db 'abc'		; felulirhato
F3	db 1,2		; nem felulirhato
F4	db 10 dup (?)		; nem felulirhato
F5	db 'a','b','c'		; nem felulirhato
S	ENDS		

Struktúra példányokat az alábbi módon tudunk előállítani:

S1	S <>		; kezdoertekek a def.-bol
S2	S <7,'ef'>		; a felulirhato ertekeket definialjuk
S3	S <,'xyz'>		; nem kell minden ertekeket felulirni

Egy struktúra mezejére például az alábbi módon tudunk hivatkozni:

```
MOV BX, OFFSET S1
```

- MOV AL, [BX].F1
- MOV AL, [BX]+F3
- MOV AL, F3.[BX]
- MOV AL, F3[BX]

Tekintsük az előző órai példát, ahol az átfedő téglalapokat vizsgáltuk, azonban a téglalapok legyenek most struktúrák. Az eljárás paramétereit, vagyis a téglalapok offszetjeit a veremben adjuk át.

```
adat      segment para public 'DATA'
teglalap  STRUC
x1        db 0                ; a bal also sarok x koordinataja
y1        db 0                ; a bal also sarok y koordinataja
x2        db 0                ; a jobb felso sarok x koordinataja
y2        db 0                ; a jobb felso sarok y koordinataja
teglalap  ENDS
t1        teglalap <2,2,4,5>
t2        teglalap <,,6,2>
adat      ends
```

```
...
MOV AX, OFFSET t1 ; az 1. teglalap offszetje
PUSH AX
MOV AX, OFFSET t2 ; az 2. teglalap offszetje
PUSH AX

CALL atfedest_vizsgal
; a visszatérési cím is bekerül a verembe
...
```

```

atfedest_vizsgal PROC
    PUSH BP                ; BP erteket menteni kell
    MOV BP, SP             ; a veremmutato erteket eltaroljuk BP-ben
    PUSH AX                ; mentjuk AX erteket
    PUSH BX                ; mentjuk BX erteket
    MOV BX, [BP+6]         ; BX = t1 offszetje
    MOV AX, [BX].x2        ; AX = t1.x2
    MOV BX, [BP+4]         ; BX = t2 offszetje
    CMP AX, [BX].x1        ; ha t1.x2 ≤ t2.x1,
    JLE nincs_atfedes     ; akkor nincs atfedes
    MOV BX, [BP+6]         ; BX = t1 offszetje
    MOV AX, [BX].x1        ; AX = t1.x1
    MOV BX, [BP+4]         ; BX = t2 offszetje
    CMP [BX].x2, AX        ; ha t2x2 ≤ t1x1,
    JLE nincs_atfedes     ; akkor nincs atfedes
    MOV BX, [BP+6]         ; BX = t1 offszetje
    MOV AX, [BX].y1        ; AX = t1.x2
    MOV BX, [BP+4]         ; BX = t2 offszetje
    CMP AX, [BX].y2        ; ha t1y1 ≥ t2y2,
    JGE nincs_atfedes     ; akkor nincs atfedes
    MOV BX, [BP+6]         ; BX = t1 offszetje
    MOV AX, [BX].y1        ; AX = t1.y1
    MOV BX, [BP+4]         ; BX = t2 offszetje
    CMP AX, [BX].y2        ; ha t2y1 ≥ t1y2,
    JGE nincs_atfedes     ; akkor nincs atfedes
    ; atfedes van
    CALL kiir_atfedo
    JMP vege                ; ugras a vegere
nincs_atfedes:
    CALL kiir_nem_atfedo
vege: POP BX                ; visszamentesek
      POP AX
      POP BP
      RET 6                ; kiuritjuk a parametereket a verembol
atfedest_vizsgal ENDP

```

(SS:SP)		BX
+2		AX
+4		BP
+6	BP+ 2	visszat. cím
+8	BP+ 4	offset t2
+10	BP+ 6	offset t1

Makrók

Olyan kódrészleteket érdemes makróban megírni, amelyeket a program során sokszor alkalmazunk. A makrók nem azonosak az eljárásokkal! A makró definíciókban megadhatjuk a makró formális paramétereit, amelyek fordítás során helyettesítődnek be.

Példa makró definícióra (32 bites összeadás):

```
MDADD  MACRO
        ADD AX, BX
        ADC DX, CX
        ENDM
```

Makró hívás: MDADD

Készítsünk olyan makrót, amely két 16 bites paramétert vár, és AX-be a két paraméterének minimumát helyezi.

```
MMIN   MACRO      P1, P2
        LOCAL vege ;; makro cimke
        PUSH BX    ;; elmentjuk BX erteket
        MOV AX, P1
        MOV BX, P2
        CMP AX, BX
        JLE vege   ;; ha AX ≤ BX, akkor kesz
        XCHG AX, BX
vege:  POP BX      ;; visszamentjuk BX-et
        ENDM
```

```
...           ; ez mar a kod
MMIN CX, DX   ; DX vagy CX tartalma a kisebb?
...           ; az kisebb ertek AX-ben van
```

Feladatok

- Adott egy **háromszög** struktúra, amelynek a három mezője a három oldalhosszat tárolja. Írj eljárást, amely eldönti, hogy létezik-e ilyen háromszög!
- Készíts makrót 16 bites előjeles egész szám abszolútértékének kiszámítására. A makró az AX regiszterében megadott érték abszolút értékét határozza meg, és az eredmény is AX-be kerüljön!