



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

34. Macroinstrucțiuni de repetare.

Macroinstrucțiunea de repetare

```
REPT <expresie_contor>  
<corpul macroinstrucțiunii >  
ENDM
```

1) Alocarea unor valori consecutive la o anumită adresă:

```
alocare      macro tabela, lungime, vi, pas  
  tabela label  byte  
  valoare = vi      ; valoare inițială  
  rept  lungime  
    valoare = valoare + pas  
    db   valoare  
  endm  
endm
```

Utilizare:

```
alocare tab1, 3, 0, 2 → tab1 db 2, 4, 6  
alocare tab2, 10, 0, 1 → tab2 db 1, . . . , 10
```

2) Generarea alfabetului, pentru litere mari, la o anumită adresă.

```
generare_alfabet  macro nume  
  nume label byte  
  i = 0  
  rept 26  
    db  'A' + i  
    i = i + 1  
  endm  
endm  
alocare      macro tab, dim  
  tab  label byte  
  i = 0  
  rept  dim  
    if   (i gt 255)  
      exitm      ;; ieșire forțată din macro  
    else      ;; înainte de execuția tuturor  
      db   i      ;; instrucțiunilor secvenței  
    endif  
    i = i + 1  
  endm  
endm
```

Macroinstrucțiuni de repetare condiționată (*Indefinite Repeat*)

```
IRP  <nume>, <<lista>>  
<corpul macroinstrucțiunii >  
ENDM
```

<nume>, va fi înlocuit, în corpul MI cu valorile din <lista>.

- 1)

```
irp    val, <1, 2, 3, 4, 5>
      dw    val*val
      endm
      dw    1, 4, 9, 16, 25
```
- 2)

```
suma  label byte
irp    x, <1, 3, 5, 7, 9>
      db    x+x
      endm
      suma db    2, 6, 10, 14, 18
```
- 3)

```
irp    reg, <ax, cx, dx, bx, sp, bp, si, di>
      push reg
      endm
```

IRPC (Indefinite RePeat Character) - *rolul valorilor numerice din IRP este preluat de caractere.*

- 1)

```
irpc  c, 12345
      dw  c*c
      endm
```
- 2) Memorarea, la o adresă a unei valori în zecimal neîmpachetat:

```
scrie_val macro adresa, valoare
      lea si, adresa
      irpc v, valoare
          mov byte ptr [si], v
          inc si
      endm
endm
```

Pentru o utilizare de forma:

```
scrie_val    adr1, 1997
```

se va genera secvența, de cod, care va depune 1, 9, 9, 7, în zec.

Dacă dorim să depunem o anumită secvență de caractere ASCII, se modifică doar linia:

```
mov byte ptr [si], '&v'
```

Utilizarea operatorilor &, % și !

Operatorul & este utilizat pentru substituirea simbolurilor, pe care le precede, cu valoarea numerică sau subșirul de caractere asociat simbolului respectiv (sau, altfel spus, concatenează valoarea asociată simbolului), în cazul utilizării simbolurilor respective ca argumente (parametrii) ale utilizării unei macroinstrucțiuni.

- 1) Declararea unor mesaje, iterative, în cadrul unui program:

```

mesaj macro n
    msg&n db 'mesaj nr.&n'
endm
    mesaj 5
va genera:    msg5 db 'mesaj nr.5'
    mesaj 11
va genera:    msg11 db 'mesaj nr.11'

```

Dacă MI are 2 parametri sau mai mulți, atunci ea se definește astfel:

```

meserr macro n, o
    err&n&o db 'eroare &n&o'
endm

```

Utilizarea acestei macro de forma: `meserr 2,2`

va genera: `err22 db 'eroare 22'`

Dublarea caracterului &, a amânat evaluarea șirului &&o, pentru momentul evaluării MI IRP.

2) Compararea reg. AL cu caracterele 'a'..'f', și execuția pentru fiecare dintre ele unei anumite secvențe (eticha, etichb, ..., etichf):

```

irpc car, abcdef
    cmp al, '&car'
    jz etich&car
endm

```

3) Definirea unei MI pentru rotirea conținutului unei destinații, cu un număr de biți și o direcție, specificate ca parametrii:

```

rotire macro sens, nr_rot, dest
    push cx ; salvare CX ; NU salv/reface CX
    mov cl, nr_rot ; sau rept nr_rot
    ro&sens dest, cl ; ro&sens dest, 1
    pop cx ; refacere CX ; endm
endm ; sau ro&sens dest, nr_rot

```

Operatorul % realizează conversia simbolului care urmează, de la o valoare numerică la un șir de caractere, ce poate fi utilizat pentru a substitui un argument, pentru un apel de MI. Acest operator, de expansiune (%), forțează evaluarea imediată a expresiei ce urmează și convertește rezultatul expresiei la un șir de cifre.

```

mesaj5 macro nume,val
    db '&nume = &val'
endm

```

și declarația:

```

const1 equ <LOW>

```

pentru a defini mesajul

```

db 'const1 = LOW'

```

se va utiliza MI anterioară, cu următorii parametrii:

mesaj5 const1, %const1

Parantezele unghiulare < > se utilizează pentru a specifica că textul dintre ele este un singur șir. Aceste paranteze se utilizează și când se dorește transmiterea unui parametru actual, pentru un apel intern de macro (apelul unei macro în altă macro).

Generarea de mesaje multiple, care respectă o anumită regulă:

```
meserr macro n
    err&n db    'eroare &n', 0Dh, 0Ah, '$'
endm
diferr macro x
    m = 0
    rept    x
           m = m + 1
           meserr %m
endm
endm
```

Un apel al acestei macroinstrucțiuni, de forma:

```
diferr 3
va      genera mesajele:
err1   db    'eroare 1', 0Dh, 0Ah, '$'
err2   db    'eroare 2', 0Dh, 0Ah, '$'
err3   db    'eroare 3', 0Dh, 0Ah, '$'
```

Un exemplu de utilizare pentru parantezele unghiulare:

```
mesaj7 macro sir
    db    '&sir',0 ;; definire șir de caractere ASCIIZ
endm
```

```
apelul:    mesaj7    <sir caractere>
va genera      db    'sir caractere',0
```

Alt operator important pentru macroinstrucțiuni este operatorul literal, semnul exclamării (!). Acest simbol comandă asamblorului ignore interpretarea caracterului următor. Acest simbol poate fi utilizat atunci când este nevoie ca unul din următoarele caractere să fie inclus într-un șir de caractere, într-o MI: ! & > %

Dacă dorim să transmitem ca argument pentru o MI un șir de caractere ce conține și caracterul '>', atunci el va fi precedat de caracterul '!', pentru a evita interpretarea lui '>' ca sfârșit de șir.

```
mesaj7    <a!>b>
care va genera mesajul
db    'a>b',0
```