ΕΝΤΟΛΕΣ ΔΙΑΚΛΑΔΩΣΗΣ - LOOP

Η LOOP μειώνει κατά 1 τον CX και εάν η τιμή του δεν είναι 0 πραγματοποιείται η διακλάδωση, αλλιώς εκτελείται η εντολή που βρίσκεται μετά την LOOP π.χ.

```
mhnhma     DB 'ΤΕΛΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ$'
..
   mov cx,18          ; μετρητής επανάληψης
   lea bx,mhnhma      ; ενεργός δ/νση ή μετατόπιση της μεταβλητής
                      ; mhnhma στον καταχωρητή δείκτη bx
loop-ektyp:
   mov dl,[bx]        ; στον dl ο χαρακτήρας προς εμφάνιση
   inc bx             ; αύξηση του δείκτη
   mov ah,02h         ; κλήση του DOS για εμφάνιση χαρακτήρα
   int 21h            ; διακοπή του DOS
   loop loop-ektyp ; επανάληψη αν ο CX > 0
```

ΕΝΤΟΛΕΣ ΔΙΑΚΛΑΔΩΣΗΣ LOOPE, LOOPNE

Η LOOPE πραγματοποιεί την επανάληψη όταν CX<>0 και ZF = 1 (δηλαδή αποτέλεσμα προηγούμενης αριθμητικής πράξης ή σύγκρισης ίσο με μηδέν).

Η **LOOPNE** πραγματοποιεί την επανάληψη όταν CX<>0 και ZF = 0 (δηλαδή αποτέλεσμα προηγούμενης αριθμητικής πράξης ή σύγκρισης διάφορο του μηδενός).

```
Buffer db 126 dup (0) ; Δήλωση προσωρινής μνήμης
..
   mov cx,256     ; μετρητής επανάληψης
   lea bx,buffer  ; ενεργός δ/νση της προσωρινής μνήμης στον καταχωρητή δείκτη bx
again:
    mov ah,8      ; κλήση του DOS για εισαγωγή χαρακτήρα από το πληκτρολόγιο
   int 21h        ; διακοπή του DOS
   mov [bx],al    ; ο χαρακτήρας που έχει εισαχθεί από την προηγούμενη κλήση του
                  ;DOS αποθηκεύεται στον AL και μεταφέρεται στην μνήμη
   inc bx         ; αύξηση του δείκτη
   cmp al,0dh     ; έλεγχος αν πατήθηκε το πλήκτρο ENTER
   loopne again   ; επανάληψη αν ο CX > 0 και ZF = 0
```

ΕΝΤΟΛΕΣ ΔΙΑΚΛΑΔΩΣΗΣ JCXZ

Τελευταία αναφέρουμε την **JCXZ** η οποία διακλαδώνεται αν ο CX = 0. Είναι ιδιαίτερα χρήσιμη πριν ξεκινήσει να εκτελείται ο βρόγχος επανάληψης όταν η τιμή του CX δεν είναι σταθερή. Π.χ

```
    JCXZ next; Αν ο CX=0 να μην εκτελεστεί η καθυστέρηση
again:
    loop again; Καθυστέρηση CX κύκλους
next:
```

Δήλωση Πίνακα

Όταν θέλουμε να δεσμεύσουμε κάποιο πλήθος θέσων στη μνήμη, στο τμήμα δεδομένων δηλώνουμε :

```
Pinakas DB 20 dup(0)
```

Με αυτή τη δήλωση δεσμεύουμε 20 bytes στη μνήμη, τα οποία τα μηδενίζουμε και το λογικό όνομα της πρώτης θέσης του πίνακα είναι Pinakas.

Όταν θέλουμε να δηλώουμε ένα πίνακα με τα περιεχόμενά του στο τμήμα δεδομένων βάζουμε:

```
Pinakas DB 23,64,77,4,109,17
```

ΔΙΕΥΘΥΝΣΙΟΠΟΙΗΣΗ

- Με την διευθυνσιοποίηση επιδιώκεται η ανάκτηση ή η ανεύρεση ενός δεδομένου στη μνήμη.
- Η προσπέλαση της μνήμης γίνεται με τον συνδυασμό δύο καταχωρητών. Ο πρώτος είναι καταχωρητής τμήματος και ο δεύτερος είναι η σταθερά ή το περιεχόμενο ενός καταχωρητή δείκτη.

ΔΙΕΥΘΥΝΣΙΟΠΟΙΗΣΗ

ΑΜΕΣΗ

```
  MOV AL,3
  MOV AL,BL
…
Data segment
Minima db ‘TELOS$’
Arit db 10,3,5,7
…
  MOV AL,Arit[0];ή MOV AL,[Arit] ή MOV
  AL,Arit
  ADD AL,Arit[1];ή ADD AL,[Arit+1]
```

ΔΙΕΥΘΥΝΣΙΟΠΟΙΗΣΗ

ΕΜΜΕΣΗ

Στην έμμεση διευθυνσιοποίηση μπορούμε να έχουμε δείκτες μνήμης μεταβλητές και όχι σταθερές όπως έχουμε στην άμεση. Ως μεταβλητές εννοούμε τα περιεχόμενα κάποιων καταχωρητών.

Με τον τρόπο αυτό και με τη βοήθεια βρόγχων μπορούμε να επεξεργαζόμαστε μεγάλα τμήματα μνήμης με πολύ λίγες εντολές.

ΔΙΕΥΘΥΝΣΙΟΠΟΙΗΣΗ

ΕΜΜΕΣΗ

Η γενική μορφή εντολών του δείκτη των τελεστών μνήμης είναι:

<καταχ. βάσης>+<καταχ. δείκτης>+<σταθερά>

Όπου καταχωρητής βάσης ένας ή κανένας από τους BX και BP, ποτέ και οι δύο.

Όπου καταχωρητής δείκτης ένας ή κανένας από τους SI και DI, ποτέ και οι δύο.

Όπου σταθερά εννοείται ένας αριθμός.

ΔΙΕΥΘΥΝΣΙΟΠΟΙΗΣΗ

ΕΜΜΕΣΗ

Ο τύπος που χρησιμοποιούμε για να σχηματίσουμε μια διεύθυνση της μνήμης είναι:

BX		SI		
ή	+	ή	+	μετατόπιση
BP		DI		

Επομένως έχουμε πολλούς συνδυασμούς