ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΣΕΡΡΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΕΠΙΚΟΙΝΩΝΙΩΝ
ΜΑΘΗΜΑ : ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ
4ο ΕΞΑΜΗΝΟ
Μαδεμλής Ιωάννης

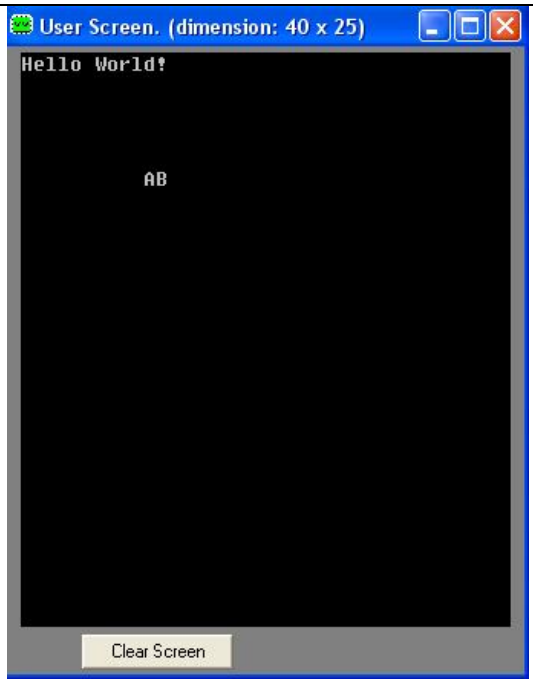
ΧΡΗΣΗ ΠΛΗΚΤΡΟΛΟΓΙΟΥ, ΟΘΟΝΗΣ ΚΑΙ INTERRUPTS ΣΤΟΝ EMULATOR

1. ΧΡΗΣΗ ΟΘΟΝΗΣ ΚΑΙ ΠΛΗΚΤΡΟΛΟΓΙΟΥ

Ο ευκολότερος τρόπος για να χρησιμοποιήσει κάποιος το πληκτρολόγιο και την οθόνη για εισαγωγή και εμφάνιση δεδομένων, αντίστοιχα, στον emulator είναι η χρήση της βιβλιοθήκης **emu8086.inc** που τον συνοδεύει. Το αρχείο αυτό πρέπει να βρίσκεται στον ίδιο κατάλογο με το αρχείο του πηγαίου κώδικα ή στον υποκατάλογο INC του emulator και περιέχει μια σειρά από μακροεντολές και procedures που χρησιμοποιούν το πληκτρολόγιο και την οθόνη του υπολογιστή. Για να χρησιμοποιηθούν οι μακροεντολές ή οι procedures της βιβλιοθήκης θα πρέπει οπωσδήποτε να υπάρχει η δήλωση **include emu8086.inc** στην αρχή του πηγαίου κώδικα ενώ μέσα στον κώδικα να υπάρχουν κλήσεις των μακροεντολών και υπορουτίνων.

Στο παράδειγμα που ακολουθεί φαίνεται η κλήση μακροεντολών της βιβλιοθήκης για εμφάνιση string και χαρακτήρων στην οθόνη:

```
include emu8086.inc

PRINT 'Hello World!'

GOTOXY 10, 5

PUTC 65          ; 65 - is an ASCII code for 'A'
PUTC 'B'

HLT

END              ; directive to stop the compiler.
```



Οι μακροεντολές που περιέχει η βιβλιοθήκη emu8086.inc είναι:

- **PUTC char** – εκτυπώνει τον χαρακτήρα (ή κωδικό ASCII) char στην τρέχουσα θέση του cursor.
- **GOTOXY col, row** – Ορίζει την θέση του cursor σε γραμμή και στήλη της οθόνης.
- **PRINT string** – Εκτυπώνει στην οθόνη ένα string.

- **PRINTN string** - Εκτυπώνει στην οθόνη ένα string όπως η προηγούμενη μακροεντολή και επιπλέον μετακινεί τον cursor στην αρχή της επόμενης γραμμής.
- **CURSOROFF** – Εξαφανίζει τον cursor από την οθόνη.
- **CURSORON** – Εμφανίζει τον cursor στην οθόνη

Οι procedures που περιέχει η βιβλιοθήκη emu8086.inc (οι οποίες καλούνται με CALL) είναι:

- **PRINT_STRING** - Procedure που εκτυπώνει στην οθόνη ένα string, τερματιζόμενο με τον χαρακτήρα ASCII 0, στην τρέχουσα θέση του cursor. Η διεύθυνση αρχής του string βρίσκεται στους καταχωρητές **DS:SI**. Για να χρησιμοποιηθεί πρέπει οπωσδήποτε να υπάρχει η δήλωση **DEFINE_PRINT_STRING** πριν την ντιρεκτίβα **END**.

- **PTHIS** - Procedure που εκτυπώνει στην οθόνη ένα string, τερματιζόμενο με τον χαρακτήρα ASCII 0 (ακριβώς όπως η PRINT_STRING), αλλά διαβάζει τη διεύθυνση του string από το σωρό. Το string πρέπει να ορίζεται ακριβώς μετά την εντολή CALL, για παράδειγμα:

  ```
  CALL PTHIS
  db 'Hello World!', 0
  ```

  Για να χρησιμοποιηθεί πρέπει οπωσδήποτε να υπάρχει η δήλωση **DEFINE_PTHIS** πριν την ντιρεκτίβα **END**.

- **GET_STRING** – Procedure που διαβάζει ένα string, τερματιζόμενο με τον χαρακτήρα ASCII 0, από το πληκτρολόγιο. Το εισαγόμενο string αποθηκεύεται στη διεύθυνση **DS:DI** και το μέγιστο μήκος του στον καταχωρητή **DX**. Η Procedure σταματά όταν πατηθεί το πλήκτρο 'Enter'. Για να χρησιμοποιηθεί πρέπει οπωσδήποτε να υπάρχει η δήλωση **DEFINE_GET_STRING** πριν την ντιρεκτίβα **END**.

- **CLEAR_SCREEN** - Procedure που καθαρίζει την οθόνη, (μέσω κύλισης όλου του παραθύρου), και θέτει τον cursor στην αρχή της. Για να χρησιμοποιηθεί πρέπει οπωσδήποτε να υπάρχει η δήλωση **DEFINE_CLEAR_SCREEN** πριν την ντιρεκτίβα **END**.

- **SCAN_NUM** - Procedure που διαβάζει από το πληκτρολόγιο έναν προσημασμένο αριθμό και τον αποθηκεύει στον καταχωρητή **CX**. Για να χρησιμοποιηθεί πρέπει οπωσδήποτε να υπάρχει η δήλωση **DEFINE_SCAN_NUM** πριν την ντιρεκτίβα **END**.

- **PRINT_NUM** - Procedure που εκτυπώνει στην οθόνη έναν προσημασμένο αριθμό που βρίσκεται στον καταχωρητή **AX**. Για να χρησιμοποιηθεί πρέπει οπωσδήποτε να υπάρχουν οι δηλώσεις **DEFINE_PRINT_NUM** και **DEFINE_PRINT_NUM_UNS** πριν την ντιρεκτίβα **END**.

- **PRINT_NUM_UNS** - Procedure που εκτυπώνει στην οθόνη έναν μη προσημασμένο αριθμό που βρίσκεται στον καταχωρητή **AX**. Για να χρησιμοποιηθεί πρέπει οπωσδήποτε να υπάρχει η δήλωση **DEFINE_PRINT_NUM_UNS** πριν την ντιρεκτίβα **END**.

Ακολουθούν παραδείγματα χρήσης των procedures:

```
#make_BIN#

include 'emu8086.inc'

;Παράδειγμα εισαγωγής και εκτύπωσης ενός string


PRINT "Doste ena string="

MOV DX,0Bh  ;το μέγιστο επιτρεπτό μήκος του string

CALL GET_STRING

PRINTN ""

PRINT "Dosate to string="

MOV SI,DI

CALL print_string

HLT

DEFINE_PRINT_STRING
DEFINE_GET_STRING

END
```

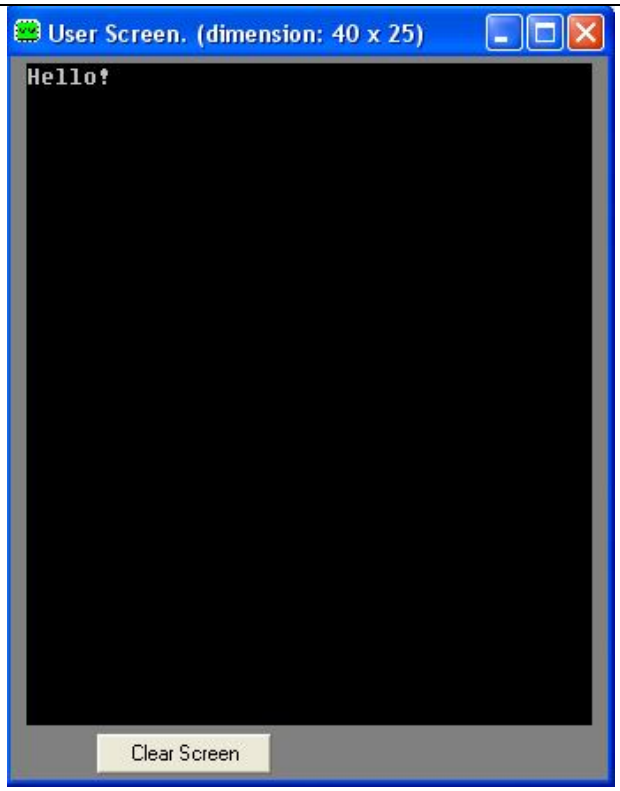

```
#make_BIN#

include 'emu8086.inc'

PRINT 'Δώστε έναν αριθμό:'

CALL   scan_num       ; διάβασε αριθμό στον CX

MOV    AX, CX         ; αντέγραψε τον στον AX

PRINTN ""

PRINT 'Δώσατε τον αριθμό:'

CALL   print_num      ; τύπωσε αριθμό στον AX

HLT

DEFINE_SCAN_NUM

DEFINE_PRINT_NUM

DEFINE_PRINT_NUM_UNS

END
```

2. ΧΡΗΣΗ INTERRUPTS ΣΤΟΝ EMULATOR

Ο emulator υποστηρίζει μια σειρά από software interrupts που είναι συμβατά με όλους τους επεξεργαστές της σειράς x86 της Intel. Τα interrupt αυτά καλούνται με την εντολή **INT** και τον αριθμό του interrupt ενώ συνήθως η τιμή του καταχωρητή **AH** καθορίζει την υπο-λειτουργία του interrupt. Σε ορισμένα interrupts χρησιμοποιούνται και άλλοι καταχωρητές για πέρασμα παραμέτρων.

Στο παράδειγμα που ακολουθεί φαίνεται η χρήση interrupts για την εκτύπωση, χαρακτήρα-χαρακτήρα, ενός string στην οθόνη:

```
#MAKE_BIN#

MOV    AH, 0Eh    ; επιλογή υπό-λειτουργίας

; INT 10h / 0Eh sub-function
; Interrupt 10h/Υπολειτουργία 0Eh
; δέχεται στον AL τον ASCII κωδικό του
; χαρακτήρα που θα εκτυπωθεί
; στην οθόνη.

MOV    AL, 'H'    ; ASCII code: 72
INT    10h        ; εκτύπωση

MOV    AL, 'e'    ; ASCII code: 101
INT    10h        ; εκτύπωση

MOV    AL, 'l'    ; ASCII code: 108
INT    10h        ; εκτύπωση

MOV    AL, 'l'    ; ASCII code: 108
INT    10h        ; εκτύπωση

MOV    AL, 'o'    ; ASCII code: 111
INT    10h        ; εκτύπωση

MOV    AL, '!'    ; ASCII code: 33
INT    10h        ; εκτύπωση

HLT
```



Τα υποστηριζόμενα interrupts συνοπτικά στην έκδοση **4.08** του emulator είναι:

INT 10h/00h
INT 10h/01h
INT 10h/02h
INT 10h/03h
INT 10h/05h
INT 10h/06h
INT 10h/07h
INT 10h/08h
INT 10h/09h
INT 10h/0Ah
INT 10h/0Ch
INT 10h/0Dh
INT 10h/0Eh
INT 10h/13h

INT 10h/1003h
INT 11h
INT 12h
INT 13h/00h
INT 13h/02h
INT 13h/03h
INT 15h/86h
INT 16h/00h
INT 16h/01h
INT 19h
INT 1Ah/00h
INT 20h

INT 21h
INT 21h/01h
INT 21h/02h
INT 21h/05h
INT 21h/06h
INT 21h/07h
INT 21h/09h
INT 21h/0Ah
INT 21h/0Bh
INT 21h/0Ch
INT 21h/0Eh
INT 21h/19h
INT 21h/25h
INT 21h/2Ah
INT 21h/2Ch

INT 21h/35h
INT 21h/39h
INT 21h/3Ah
INT 21h/3Bh
INT 21h/3Ch
INT 21h/3Dh
INT 21h/3Eh
INT 21h/3Fh
INT 21h/40h
INT 21h/41h
INT 21h/42h
INT 21h/47h
INT 21h/4Ch
INT 21h/56h

INT 33h/0000h
INT 33h/0001h
INT 33h/0002h
INT 33h/0003h

Ακολουθεί αναλυτική περιγραφή κάθε κατηγορίας interrupt:

3. BIOS INTERRUPTS ΣΤΟΝ EMULATOR

▶ INT 10h / AH = 0 – ορίζει το video mode.

Είσοδος **AL** = επιθυμητό video mode. υποστηρίζονται τα ακόλουθαvideo modes:
00h - text mode. 40x25. 16 χρώματα. 8 σελίδες.
03h - text mode. 80x25. 16 χρώματα. 8 σελίδες.
13h - graphical mode. 40x25. 256 χρώματα. 320x200 pixels. 1 σελίδα.

Παράδειγμα:

```
mov al, 13h
mov ah, 0
int 10h
```

▶ INT 10h / AH = 01h – Ορίζει το σχήμα του cursor σε text-mode.

Είσοδος **CH** = γραμμή αρχής του cursor (bits 0-4) και επιλογές (bits 5-7). **CL** = κάτω γραμμή του cursor (bits 0-4). Όταν το bit 5 του CH είναι **0**, ο cursor είναι ορατός. Όταν το bit 5 είναι **1**, ο cursor είναι αόρατος.

```
; hide blinking text cursor:
        mov ch, 32
        mov ah, 1
        int 10h

; show standard blinking text cursor:
        mov ch, 6
        mov cl, 7
        mov ah, 1
        int 10h

; show box-shaped blinking text cursor:
        mov ch, 0
        mov cl, 7
        mov ah, 1
        int 10h

;       note: some bioses required CL to be >=7,
;       otherwise wrong cursor shapes are displayed.
```

▶ INT 10h / AH = 2 – Ορίζει τη θέση του cursor στην οθόνη.

Είσοδος **DH** = γραμμή. **DL** = στήλη. **BH** = αριθμός σελίδος (0..7).

Παράδειγμα:

```
mov dh, 10
mov dl, 20
mov bh, 0
mov ah, 2
int 10h
```

▶ INT 10h / AH = 03h – Επιστρέφει τη θέση και το μέγεθος του cursor.

Είσοδος: **BH** = αριθμός σελίδας.

Επιστρέφει: **DH** = γραμμή cursor. **DL** = στήλη cursor. **CH** = γραμμή έναρξης cursor. **CL** = κάτω γραμμή cursor.

INT 10h / AH = 05h – Επιλέγει την ενεργή σελίδα video.

Είσοδος: **AL** = αριθμός νέας σελίδας video (0..7).

Εμφανίζεται η ενεργή σελίδα

INT 10h / AH = 06h – Ολίσθηση παραθύρου προς τα πάνω.
INT 10h / AH = 07h - Ολίσθηση παραθύρου προς τα κάτω.

Είσοδος:
AL = αριθμός γραμμών ολίσθησης (00h = καθαρισμός όλου του παραθύρου).
BH = ιδιότητα (attribute) που χρησιμοποιείται για εγγραφή κενών γραμμών στο κάτω μέρος του παραθύρου.
CH, CL = γραμμή, στήλη της άνω, αριστερής γωνίας του παραθύρου.
DH, DL = γραμμή, στήλη της κάτω, δεξιάς γωνίας του παραθύρου.

INT 10h / AH = 08h – Διάβασε χαρακτήρα και ιδιότητες στην τρέχουσα θέση του cursor.

*Είσοδος:***BH** = αριθμός σελίδας.

*Επιστρέφει:***AH** = ιδιότητες. **AL** = χαρακτήρας.

INT 10h / AH = 09h – Εκτυπώνει χαρακτήρα και ιδιότητες στην τρέχουσα θέση του cursor.

*Είσοδος:***AL** = χαρακτήρας που θα εκτυπωθεί. **BH** = αριθμός σελίδας. **BL** = Ιδιότητες. **CX** = πόσες φορές θα εκτυπωθεί ο χαρακτήρας.

INT 10h / AH = 0Ah – Εκτύπωση μόνον χαρακτήρα στην τρέχουσα θέση του cursor.

*Είσοδος:***AL** = χαρακτήρας που θα εκτυπωθεί. **BH** = αριθμός σελίδας. **CX** = πόσες φορές θα εκτυπωθεί ο χαρακτήρας.

INT 10h / AH = 0Ch – Αλλαγή χρώματος ενός μοναδικού pixel.

*Είσοδος:***AL** = χρώμα color. **CX** = στήλη. **DX** = γραμμή.

Παράδειγμα:

```
mov al, 13h
mov ah, 0
int 10h     ; set graphics video mode.
mov al, 1100b
mov cx, 10
mov dx, 20
mov ah, 0ch
int 10h     ; set pixel.
```

INT 10h / AH = 0Dh – Επιστρέφει το χρώμα ενός μοναδικού pixel.

Είσοδος: **CX** = στήλη. **DX** = γραμμή.

Έξοδος: **AL** = χρώμα pixel

INT 10h / AH = 0Eh – Εκτύπωση χαρακτήρα σαν τηλέτυπο.

Είσοδος: **AL** = χαρακτήρας προς εκτύπωση.

Εκτυπώνει έναν χαρακτήρα στην οθόνη, προχωρά τον cursor και ολισθαίνει την οθόνη όσο χρειάζεται, η εκτύπωση γίνεται πάντα στην ενεργή σελίδα video.

Παράδειγμα:

```
mov al, 'a'
mov ah, 0eh
int 10h

; note: on specific systems this
; function may not be supported in graphics mode.
```

INT 10h / AH = 13h - Εκτύπωση string.

Είσοδος:
AL = Τρόπος εγγραφής:
bit 0: ενημέρωση cursor μετά την εγγραφή;
bit 1: το string περιέχει ιδιότητες.
BH = αριθμός σελίδας.
BL = Ιδιότητες αν το string περιέχει μόνο χαρακτήρες (το bit 1 του AL είναι 0).
CX = αριθμός χαρακτήρων του string (δεν μετριούνται οι ιδιότητες).
DL,DH = στήλη, γραμμή της οθόνης που αρχίζει η εκτύπωση του string.
ES:BP δείκτες στη διεύθυνση στη μνήμη που βρίσκεται το προς εκτύπωση string.

Παράδειγμα:

```
mov al, 1
mov bh, 0
mov bl, 0011_1011b
mov cx, msg1end - offset msg1 ; calculate message size.
mov dl, 10
mov dh, 7
push cs
pop es
mov bp, offset msg1
mov ah, 13h
int 10h
jmp msg1end
msg1 db " hello, world! "
msg1end:
```

INT 10h / AX = 1003h – Εναλλαγή έντονων χαρακτήρων/αναβοσβησίματος χαρακτήρων.

Είσοδος:
BL = Τρόπος εγγραφής:
0: ενεργοποίηση έντονων χαρακτήρων.
1: ενεργοποίηση αναβοσβησίματος χαρακτήρων (δεν υποστηρίζεται από τον emulator)
BH = 0 (προς αποφυγή προβλημάτων σε ορισμένες κάρτες γραφικών).

Παράδειγμα:

```
mov ax, 1003h
mov bx, 0
int 10h
```

Πίνακας χρωμάτων bit:
Οι ιδιότητες ενός χαρακτήρα είναι μια 8 bit τιμή, τα 4 bit χαμηλής τάξης καθορίζουν το χρώμα του χαρακτήρα και τα 4 bit υψηλής τάξης το χρώμα υποβάθρου του χαρακτήρα. Σημείωση: Αν και ο emulator και η γραμμή εντολών των Windows δεν υποστηρίζουν αναβόσβημα του υποβάθρου των χαρακτήρων, ωστόσο συνίσταται να απενεργοποιείται το αναβόσβημα υποβάθρου από τον προγραμματιστή.

HEX	BIN	COLOR
0	0000	black
1	0001	blue
2	0010	green
3	0011	cyan
4	0100	red
5	0101	magenta
6	0110	brown
7	0111	light gray
8	1000	dark gray
9	1001	light blue
A	1010	light green
B	1011	light cyan
C	1100	light red
D	1101	light magenta
E	1110	yellow
F	1111	white

Έτσι, π.χ., η τιμή ιδιότητας (attribute) 0Eh αντιστοιχεί σε κίτρινο χαρακτήρα σε μαύρο υπόβαθρο ενώ η τιμή 4Fh αντιστοιχεί σε λευκό χαρακτήρα σε κόκκινο υπόβαθρο.

Σημείωση: Ο παρακάτω κώδικας χρησιμοποιείται για συμβατότητα με την εμφάνιση πλήρους οθόνης της γραμμής εντολών του DOS:

```
; use this code for compatibility with dos/cmd prompt full screen mode:
mov     ax, 1003h
mov     bx, 0   ; disable blinking.
int     10h
```

▶ INT 11h – Ανάγνωση της λίστας εξοπλισμού από το BIOS.

Επιστρέφει: **AX** = 16-bit τιμή του εξοπλισμού του Η/Υ από το BIOS, στην ουσία το interrupt αυτό επιστρέφει τα περιεχόμενα της θέσης μνήμης 0040h:0010h και της επόμενης.

Προς το παρών, στον emulator επιστρέφει μόνο τον αριθμό των εγκατεστημένων οδηγών δισκετών.

Σημασία των bit που επιστρέφει το BIOS:

bit(s)	Σημασία
15-14	Αριθμός των παράλληλων θυρών.
13	Δεσμευμένο.
12	Υπάρχει Game port.
11-9	Αριθμός σειριακών θυρών.
8	Δεσμευμένο.
7-6	Αριθμός οδηγών δισκετών (μείον 1): 00 ένας οδηγός; 01 δύο οδηγοί; 10 τρεις οδηγοί; 11 τέσσερις οδηγοί.
5-4	Αρχικό video mode: 00 EGA,VGA,PGA, ή άλλη κάρτα με ενσωματωμένο video BIOS; 01 40x25 CGA έγχρωμη.

	10 80x25 CGA έγχρωμη (emulator default). 11 80x25 μονόχρωμο κείμενο.
3	Δεσμευμένο.
2	Υπάρχει ποντίκι PS/2.
1	Υπάρχει μαθηματικός συνεπεξεργαστής.
0	1 όταν έχει επιλεγεί εκκίνηση από δισκέτα.

▶ **INT 12h** – **Επιστρέφει το μέγεθος της μνήμης**.

*Επιστρέφει:***AX** = Τα kilobytes συνεχούς μνήμης, ξεκινώντας από την απόλυτη διεύθυνση 00000h. Το interrupt αυτό επιστρέφει τα περιεχόμενα της θέσης μνήμης 0040h:0013h και της επόμενης.

Για τα ακόλουθα interrupts οι μονάδες δισκέτας προσομοιώνονται μέσω των αρχείων **FLOPPY_0** (..3).

▶ **INT 13h** / **AH = 00h** – **Εκτελεί επαναφορά (reset) του συστήματος δίσκων**. (στην παρούσα έκδοση του emulator το interrupt αυτό δεν λειτουργεί)

▶ **INT 13h** / **AH = 02h** – **Διαβάζει sectors από το δίσκο και τα αντιγράφει στη μνήμη**.
▶ **INT 13h** / **AH = 03h** – **Γράφει sectors στο δίσκο**.

Είσοδος:
AL = αριθμός των sectors που θα διαβαστούν/εγγραφούν (πρέπει να είναι ≠ 0)
CH = αριθμός κυλίνδρου (0..79).
CL = αριθμός sector (1..18).
DH = αριθμός κεφαλής (0..1).
DL = αριθμός δίσκου (0..3 , για τον emulator εξαρτάται από τον αριθμό των FLOPPY_ files).
ES:BX δείκτης στα δεδομένα.

Επιστρέφει:
CF γίνεται 1 αν υπάρχει σφάλμα.
CF γίνεται 0 αν η εγγραφή/ανάγνωση είναι επιτυχής.
AH = κατάσταση (0 – αν είναι επιτυχής).
AL = αριθμός sectors που μεταφέρθηκαν.

Σημείωση: κάθε sector έχει **512** bytes.

▶ **INT 15h** / **AH = 86h** – **Λειτουργία αναμονής του BIOS**.

*Είσοδος:***CX:DX** = χρονικό διάστημα αναμονής σε microseconds

Επιστρέφει:
CF 0 αν η λειτουργία είναι επιτυχής (το χρονικό διάστημα αναμονής έχει συμπληρωθεί),
CF 1 αν υπάρχει σφάλμα ή το χρονικό διάστημα αναμονής δεν έχει ακόμη συμπληρωθεί.

Σημείωση:η ανάλυση της περιόδου αναμονής είναι 977 ì s σε πολλά συστήματα. Τα Windows XP δεν υποστηρίζουν το interrupt αυτό και θέτουν πάντα CF=1.

▶ **INT 16h** / **AH = 00h** – **Διαβάζει ένα πλήκτρο από το πληκτρολόγιο (χωρίς ηχώ στην οθόνη)**.

*Επιστρέφει:***AH** = ο scan code του BIOS. **AL** = ο ASCII χαρακτήρας. (αν έχει πατηθεί πλήκτρο, αυτό αφαιρείται από την buffer του πληκτρολογίου).

▶ **INT 16h / AH = 01h – Ελέγχει αν έχει πατηθεί πλήκτρο στην buffer του πληκτρολογίου**.

Επιστρέφει:
ZF = 1 αν δεν έχει πατηθεί πλήκτρο.
ZF = 0 αν έχει πατηθεί πλήκτρο.
AH = ο scan code του BIOS.
AL = ο ASCII χαρακτήρας.
(αν έχει πατηθεί πλήκτρο, αυτό δεν αφαιρείται από την buffer του πληκτρολογίου).

▶ **INT 19h – Επανεκκίνηση συστήματος**.
Συνήθως, το BIOS θα προσπαθήσει να διαβάσει το sector 1, κεφαλή 0, track 0 από τον οδηγό **A:** στη διεύθυνση 0000h:7C00h. Ο emulator σταματά την εκτέλεση του interrupt, για εκκίνηση από δισκέτα επιλέξτε από το μενού: **'virtual drive' -> 'boot from floppy'**

▶ **INT 1Ah / AH = 00h – Διαβάζει την ώρα του συστήματος**.

*Επιστρέφει:***CX:DX** = ο αριθμός χτύπων του ρολογιού από τα μεσάνυχτα. **AL** = μετρητής μεσονυχτίων, αυξάνει κατά 1 τα μεσάνυχτα κάθε ημέρας.

Σημειώσεις: Υπάρχουν περίπου **18.20648** χτύποι ρολογιού κάθε δευτερόλεπτο και **1800B0h** κάθε 24 ώρες. Ο καταχωρητής **AL** δεν παίρνει τιμή στον emulator.

▶ **INT 20h – Έξοδος στο λειτουργικό σύστημα**.

4. MS-DOS INTERRUPTS (INT 21H) ΣΤΟΝ EMULATOR

Τα interrupts του MS-DOS εκτελούνται από τον emulator σε προσομοίωση. Το σύστημα αρχείων του DOS προσομοιώνεται στον κατάλογο **C:\emu8086\vdrive\x** (x είναι το γράμμα μιας μονάδας δίσκου. Αν δεν καθορίζεται γράμμα μονάδας δίσκου και τρέχον κατάλογος τότε χρησιμοποιείται εξ ορισμού η διαδρομή **C:\emu8086\MyBuild**. Τα αρχεία **FLOPPY_0,1,2,3** προσομοιώνονται ανεξάρτητα από το DOS.

Για τον emulator το drive **A:** προσομοιώνεται με το αρχείο **c:\emu8086\FLOPPY_0** (για τα Interrupts του BIOS **INT 13h** και εκκίνησης boot). Για τα DOS interrupts (**INT 21h**) το drive **A:** προσομοιώνεται στον υποκατάλογο: **C:\emu8086\vdrive\a**

Σημείωση: Το DOS περιορίζει το μέγεθος του ονόματος αρχείων και καταλόγων στους 8 χαρακτήρες και 3 χαρακτήρες επέκταμα. Παράδειγμα έγκυρου ονόματος αρχείου είναι το: **myfile.txt** (όνομα αρχείου = 6 χαρακτήρες, επέκταμα = 3 χαρακτήρες). Το επέκταμα γράφεται μετά την τελεία και δεν επιτρέπονται άλλες τελείες.

▶ **INT 21h** / **AH=1** – Διαβάζει έναν χαρακτήρα από το πληκτρολόγιο, με εμφάνιση στην οθόνη, ο χαρακτήρας αποθηκεύεται στον καταχωρητή **AL**. Αν δεν υπάρχει χαρακτήρας στην buffer του πληκτρολογίου, αναμένει να πατηθεί πλήκτρο.

▶ **INT 21h** / **AH=2** – Εμφανίζει έναν χαρακτήρα στην οθόνη.

Είσοδος: **DL** = χαρακτήρας προς εμφάνιση, μετά την εκτέλεση του interrupt είναι **AL = DL**.

▶ **INT 21h** / **AH=5** – Εκτυπώνει ένα χαρακτήρα στον εκτυπωτή.

Είσοδος: **DL** = χαρακτήρας προς εκτύπωση, μετά την εκτέλεση του interrupt είναι **AL = DL**.

▶ **INT 21h** / **AH=6** – Απευθείας έξοδος στην οθόνη ή είσοδος από το πληκτρολόγιο.
Παράμετρος για έξοδο: **DL** = 0..254 (Κωδικός ASCII)
Παράμετρος για είσοδο: **DL** = 255
Για έξοδο επιστρέφει: **AL = DL**
Για είσοδο επιστρέφει: **ZF=1** αν δεν υπάρχει διαθέσιμος χαρακτήρας και **AL = 00h**, **ZF=0** αν υπάρχει διαθέσιμος χαρακτήρας. **AL** = ο χαρακτήρας που εισήχθη, η buffer αδειάζει.

Παράδειγμα:

```
mov ah, 6
mov dl, 'a'
int 21h      ; output character.

mov ah, 6
mov dl, 255
int 21h      ; get character from keyboard buffer (if any) or set ZF=1.
```

▶ **INT 21h** / **AH=7** – Ανάγνωση ενός χαρακτήρα από το πληκτρολόγιο, χωρίς εμφάνιση στην οθόνη, και αποθήκευση στον καταχωρητή **AL**. Αν δεν υπάρχει χαρακτήρας στην buffer του πληκτρολογίου, αναμένει να πατηθεί πλήκτρο.

▶ **INT 21h** / **AH=9** – Εμφανίζει στην οθόνη ένα string που αρχίζει στη διεύθυνση **DS:DX**. Το string πρέπει να τερματίζεται με τον χαρακτήρα '**$**'.

INT 21h / **AH=0Ah** – Διαβάζει ένα string από το πληκτρολόγιο και το αποθηκεύει ξεκινώντας από τη διεύθυνση **DS:DX**, όπου το πρώτο byte είναι το μέγεθος της buffer και το δεύτερο byte είναι οι χαρακτήρες που εισήχθησαν. Το interrupt αυτό **δεν** προσθέτει τον χαρακτήρα '**$**' στο τέλος του string. Για εμφάνιση του string στην οθόνη χρησιμοποιώντας το **INT 21h** / **AH=9** θα πρέπει να προσθέσει ο προγραμματιστής τον χαρακτήρα '**$**' στο τέλος του string και να αρχίσει την εμφάνιση του string από τη διεύθυνση **DS:DX + 2**.

INT 21h / **AH=0Bh** – Επιστρέφει την κατάσταση του πληκτρολογίου.

Επιστρέφει: **AL = 00h** εάν δεν υπάρχει διαθέσιμος χαρακτήρας, **AL = 0FFh** εάν υπάρχει διαθέσιμος χαρακτήρας.

INT 21h / **AH=0Ch** – Αδειάζει τη buffer του πληκτρολογίου και διαβάζει το πληκτρολόγιο.

Είσοδος: **AL** = ο αριθμός της συνάρτησης εισόδου που θα εκτελεστεί μετά το άδειασμα της buffer (μπορεί να είναι 01h,06h,07h,08h, ή 0Ah – για άλλες τιμές η buffer αδειάζει αλλά δεν γίνεται ανάγνωση πληκτρολογίου). Άλλοι καταχωρητές παίρνουν τιμές ανάλογα με την συνάρτηση που θα εκτελεστεί.

INT 21h / **AH= 0Eh** – Επιλέγει το default drive.

Είσοδος: **DL** = Νέο default drive (0=A:, 1=B:, κ.λ.π)

Επιστρέφει: **AL** = ο αριθμός των εν δυνάμει έγκυρων γραμμάτων drive

Σημείωση: η επιστρεφόμενη τιμή είναι τα υψηλότερο εγκατεστημένο drive.

INT 21h / **AH= 19h** – Επιστρέφει το τρέχον drive.

Επιστρέφει: **AL** = drive (0=A:, 1=B:, κ.λ.π)

INT 21h / **AH=25h** – Ορίζει το interrupt vector.

Είσοδος: **AL** = αριθμός interrupt. **DS:DX** -> η νέα διεύθυνση της ρουτίνας εξυπηρέτησης αυτού του interrupt.

INT 21h / **AH=2Ah** – Επιστρέφει την ημερομηνία συστήματος.

Επιστρέφει: **CX** = έτος (1980-2099). **DH** = μήνας. **DL** = ημέρα. **AL** = ημέρα της εβδομάδος (00h=Κυριακή)

INT 21h / **AH=2Ch** – Επιστρέφει την ώρα συστήματος.

Επιστρέφει: **CH** = ώρα. **CL** = λεπτά. **DH** = δευτερόλεπτα. **DL** = 1/100 του δευτερολέπτου.

INT 21h / **AH=35h** – Επιστρέφει την διεύθυνση της ρουτίνας εξυπηρέτησης ενός interrupt;

Είσοδος: **AL** = αριθμός interrupt.

Επιστρέφει: **ES:BX** -> τρέχουσα διεύθυνση της ρουτίνας εξυπηρέτησης αυτού του interrupt.

INT 21h / **AH= 39h** – Δημιουργεί κατάλογο.

Είσοδος: **DS:DX** -> String τερματιζόμενο με τον χαρακτήρα ASCII 0 string που δηλώνει το path

INT 21h / AH= 3Ah – Διαγράφει κατάλογο.

Είσοδος: **DS:DX** -> Το path (σε ASCII) του καταλόγου προς διαγραφή

Επιστρέφει: **CF=0** εάν διαγραφή επιτυχής, **CF=1** εάν όχι επιτυχής, **AX** = κωδικός σφάλματος.

Σημείωση: ο κατάλογος πρέπει να είναι κενός (χωρίς αρχεία στο εσωτερικό του).

INT 21h / AH= 3Bh – Ορίζει τον τρέχοντα κατάλογο.

Είσοδος: **DS:DX** -> Το path (σε ASCII) του καταλόγου που θα γίνει ενεργός (max 64 bytes).

Επιστρέφει: **CF=0** εάν επιτυχής, **CF=1** εάν όχι επιτυχής, **AX** = κωδικός σφάλματος.

Σημείωση: ακόμη και αν το όνομα του νέου καταλόγου περιλαμβάνει ένα γράμμα από drive, το τρέχον drive δεν αλλάζει, μόνο ο τρέχον κατάλογος στο drive αυτό.

INT 21h / AH= 3Ch – Δημιουργεί ή μηδενίζει ένα αρχείο.

Είσοδος: **DS:DX** -> Το όνομα του αρχείου σε ASCII.
CX = ιδιότητες αρχείου:

```
mov cx, 0     ; normal - no attributes.
mov cx, 1     ; read-only.
mov cx, 2     ; hidden.
mov cx, 4     ; system
mov cx, 7     ; hidden, system and read-only!
mov cx, 16    ; archive
```

Επιστρέφει: **CF=0** εάν επιτυχής, **AX** = file handle. **CF=1** εάν όχι επιτυχής **AX** = κωδικός σφάλματος.

Σημείωση: εάν το αρχείο υπάρχει διαγράφεται χωρίς προειδοποίηση!

Παράδειγμα:

```
org 100h
mov ah, 3ch
mov cx, 0
mov dx, offset filename
mov ah, 3ch
int 21h
jc err
mov handle, ax
jmp k
filename db "myfile.txt", 0
handle dw ?
err:
; ....
k:
ret
```

INT 21h / AH= 3Dh – Ανοίγει ένα υπάρχον αρχείο.

Είσοδος: **DS:DX** -> Το όνομα του αρχείου σε ASCII.
AL = modes προσπέλασης και διαμοιρασμού:

```
mov al, 0  ; read
mov al, 1  ; write
```

```
mov al, 2   ; read/write
```

Επιστρέφει **CF=0** εάν επιτυχής, **AX** = file handle. **CF=1** εάν όχι επιτυχής **AX** = κωδικός σφάλματος.

Σημείωση: ο δείκτης αρχείου πρέπει να είναι στην αρχή του αρχείου και το αρχείο να υπάρχει.

Παράδειγμα:

```
org 100h
mov al, 2
mov dx, offset filename
mov ah, 3dh
int 21h
jc err
mov handle, ax
jmp k
filename db "myfile.txt", 0
handle dw ?
err:
; ....
k:
ret
```

INT 21h / AH= 3Eh – Κλείνει ένα αρχείο.

Είσοδος: **BX** = file handle

Επιστρέφει: **CF=0** εάν επιτυχής, **CF=1** εάν όχι επιτυχής **AX** = κωδικός σφάλματος. (06h)

INT 21h / AH= 3Fh – Διαβάζει δεδομένα από αρχείο.

Είσοδος: **BX** = file handle. **CX** = αριθμός bytes που θα διαβαστούν. **DS:DX** -> buffer για τα δεδομένα.

Επιστρέφει: **CF=0** εάν επιτυχής - **AX** = αριθμός bytes που διαβάστηκαν πραγματικά; 0 εάν EOF (end of file) πριν την κλήση του interrupt. **CF=1** εάν όχι επιτυχής **AX** = κωδικός σφάλματος.

Σημείωση: η ανάγνωση των δεδομένων ξεκινά από την τρέχουσα θέση του αρχείου και η θέση του αρχείου ενημερώνεται μετά από μια επιτυχή ανάγνωση. Στην περίπτωση που η ανάγνωση δεν ξεκινήσει από την αρχή του αρχείου τότε η τιμή του **AX** θα είναι μικρότερη της τιμής του **CX**.

INT 21h / AH= 40h – Εκτελεί εγγραφή δεδομένων σε αρχείο.

Είσοδος: **BX** = file handle. **CX** = αριθμός bytes προς εγγραφή. **DS:DX** -> δεδομένα προς εγγραφή.

Επιστρέφει **CF=0** εάν επιτυχής - **AX** = αριθμός bytes που εγγράφηκαν πραγματικά. **CF=1** εάν όχι επιτυχής **AX** = κωδικός σφάλματος

Σημείωση: εάν **CX=0**, δεν γράφονται δεδομένα και το αρχείο μηδενίζεται. Η εγγραφή των δεδομένων γίνεται ξεκινώντας από την τρέχουσα θέση του αρχείου η οποία θέση ενημερώνεται μετά από μια επιτυχή εγγραφή. Συνήθης λόγος που **AX** < **CX** στην επιστροφή είναι γεμάτος δίσκος.

INT 21h / AH= 41h – Διαγράφει ένα αρχείο.

Είσοδος: **DS:DX** -> Το όνομα του αρχείου σε ASCII (όχι χαρακτήρες μπαλαντέρ ?,!,*).

Επιστρέφει: **CF=0** εάν επιτυχής, **AL** το drive του διαγραμμένου αρχείου (μη τεκμηριωμένη). **CF=1** εάν όχι επιτυχής **AX** = κωδικός σφάλματος.

Σημείωση: Το DOS δεν διαγράφει τα δεδομένα ενός αρχείου, απλώς αυτά γίνονται μη προσπελάσιμα λόγω ενημέρωσης του FAT. Διαγραφή ενός ανοικτού αρχείου μπορεί να προκαλέσει βλάβη στο σύστημα αρχείων.

INT 21h / AH= 42h – Λειτουργία SEEK – Ορίζει την τρέχουσα θέση αρχείου.

Είσοδος:
AL = αφετηρία της μετατόπισης: **0** – αρχή του αρχείου. **1** – τρέχουσα θέση αρχείου. **2** – τέλος αρχείου.
BX = file handle.
CX:DX = μετατόπιση από την αρχική θέση αρχείου.

Επιστρέφει: **CF=0** εάν επιτυχής, **DX:AX** = νέα θέση αρχείου σε bytes από την αρχή του αρχείου. **CF=1** εάν όχι επιτυχής **AX** = κωδικός σφάλματος.

Σημείωση: για αφετηρία μετατόπισης **1** και **2**, ο δείκτης μπορεί να τοποθετηθεί πριν την αρχή του αρχείου. Στην περίπτωση αυτή δεν υπάρχει σφάλμα αλλά θα υπάρξει σφάλμα αν κατόπιν γίνουν απόπειρες εγγραφής ή ανάγνωσης του αρχείου. Εάν η νέα θέση είναι πέρα από το τρέχον τέλος του αρχείου, το αρχείο θα επεκταθεί με την επόμενη εγγραφή.

Παράδειγμα:

```
org 100h
mov ah, 3ch
mov cx, 0
mov dx, offset filename
mov ah, 3ch
int 21h  ; create file...
mov handle, ax

mov bx, handle
mov dx, offset data
mov cx, data_size
mov ah, 40h
int 21h ; write to file...

mov al, 0
mov bx, handle
mov cx, 0
mov dx, 7
mov ah, 42h
int 21h ; seek...

mov bx, handle
mov dx, offset buffer
mov cx, 4
mov ah, 3fh
int 21h ; read from file...

mov bx, handle
mov ah, 3eh
int 21h ; close file...
ret

filename db "myfile.txt", 0
handle dw ?
data db " hello files! "
data_size=$-offset data
buffer db 4 dup(' ')
```

INT 21h / **AH= 47h** – Επιστρέφει τον τρέχοντα κατάλογο.

Είσοδος: **DL** = αριθμός drive (00h = το default, 01h = A:, κ.λ.π.). **DS:SI** -> buffer μεγέθους 64 byte για το path σε ASCII.

Επιστρέφει: **CF=0** εάν επιτυχής. **CF=1** εάν όχι επιτυχής, **AX** = κωδικός σφάλματος (0Fh)

Σημείωση: το επιστρεφόμενο path δεν περιλαμβάνει το γράμμα του drive και το αρχικό backslash.

INT 21h / **AH=4Ch** - Επιστρέφει τον έλεγχο στο λειτουργικό σύστημα.

INT 21h / **AH= 56h** – Μετονομάζει/μετακινεί ένα αρχείο.

Είσοδος: **DS:DX** -> όνομα υπάρχοντος αρχείου σε ASCII. **ES:DI** -> νέο όνομα αρχείου σε ASCII.

Επιστρέφει: **CF=0** εάν επιτυχής. **CF=1** εάν όχι επιτυχής, **AX** = κωδικός σφάλματος.

Σημείωση: επιτρέπει μετακίνηση αρχείων μεταξύ καταλόγων μόνον στο ίδιο λογικό drive. Ανοικτά αρχεία δεν πρέπει να μετονομάζονται!

5. INTERRUPTS (INT 33H) ΠΟΥ ΣΧΕΤΙΖΟΝΤΑΙ ΜΕ ΤΟ ΠΟΝΤΙΚΙ

INT 33h / **AX=0000** – Αρχικοποίηση ποντικιού. Κάθε προηγούμενος δείκτης ποντικιού αποκρύπτεται.

Επιστρέφει: εάν επιτυχής: **AX**=0FFFFh και **BX**=αριθμός πλήκτρων ποντικιού. Αν μη επιτυχής τότε **AX**=0

Παράδειγμα:

```
mov ax, 0
int 33h
```

INT 33h / **AX=0001** – Εμφανίζει τον δείκτη του ποντικιού.

Παράδειγμα:

```
mov ax, 1
int 33h
```

INT 33h / **AX=0002** – Εξαφανίζει τον δείκτη του ποντικιού.

Παράδειγμα::

```
mov ax, 2
int 33h
```

INT 33h / **AX=0003** – Επιστρέφει τη θέση του ποντικιού και την κατάσταση των πλήκτρων του

Επιστρέφει:

Αν είναι πατημένο το αριστερό κουμπί: **BX=1**
Αν είναι πατημένο το δεξιό κουμπί: **BX=2**
Αν και τα 2 πλήκτρα είναι πατημένα: **BX=3**
CX = x
DX = y

Παράδειγμα:

```
mov ax, 3
int 33h
; note: in graphical 320x200 mode the value of CX is doubled.
; see mouse2.asm in examples.
```