



**ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

---

## **Αρχιτεκτονική Υπολογιστών**

### **Ασκήσεις Εργαστηρίου**

**Ενότητα: ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ Νο 10**

Δρ. Μηνάς Δασυγένης

[mdasyg@ieee.org](mailto:mdasyg@ieee.org)

**Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών**

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

[http:// arch.icte.uowm.gr/mdasyg](http://arch.icte.uowm.gr/mdasyg)

---

## Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



## Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα του Πανεπιστημίου Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

## Περιεχόμενα

1.	Σκοπός της άσκησης .....	4
2.	Εργαστηριακή Άσκηση .....	4
3.	Συναρτήσεις .....	4
3.1	setgraphicsmode .....	4
3.2	putpixel .....	4
3.3	drawline1 .....	5
3.4	drawline_horizontal_right.....	6
3.5	drawline_horizontal_left.....	7
3.6	drawline_vertical_top.....	7
3.7	drawline_vertical_bottom.....	7
3.8	Συνάρτηση χειρισμού εξαιρέσεων .....	7
3.9	Συνάρτηση χειρισμού διακοπής για περιφερειακό .....	9
4.	Ερωτήσεις κατανόησης .....	10
5.	Βοήθεια με Ερωτήσεις και Απαντήσεις.....	11

## 1. Σκοπός της άσκησης

- Είσοδος Έξοδος με χαρτογράφηση μνήμης.Γραφικά στην assembly x86.
- Τοποθέτηση pixel.
- Σχεδιασμός κατακόρυφων γραμμών.
- Συναρτήσεις χειρισμού εξαιρέσεων/διακοπών.

**(A) 7 Ερωτήσεις**

**(C) 9 Ασκήσεις**

## 2. Εργαστηριακή Άσκηση

Σε αυτό το εργαστήριο θα δούμε την επικοινωνία του επεξεργαστή με την κάρτα γραφικών. Η συγκεκριμένη επικοινωνία επιτυγχάνεται με την εγγραφή ή ανάγνωση των διευθύνσεων μνήμης που έχει απεικονιστεί η μνήμη της κάρτας γραφικών (*είσοδος-έξοδος με χαρτογράφηση μνήμης*). Έτσι, αν ο επεξεργαστής θέλει να γράψει ένα pixel στην οθόνη, αρκεί να υπολογίσει τη διεύθυνση στη μνήμη για το συγκεκριμένο pixel και στη συνέχεια να μεταφέρει μια τιμή που αντιστοιχεί στο συγκεκριμένο χρώμα σε αυτή τη διεύθυνση. Ομοίως, αν θέλει να διαβάσει το χρώμα ενός pixel πρέπει να μεταφέρει από τη δ/ση μνήμης κάρτας γραφικών που αντιστοιχεί στο συγκεκριμένο pixel, την τιμή του pixel σε έναν καταχωρητή. Αν και η κάρτα γραφικών έχει δικιά της μνήμη, εντούτοις η επικοινωνία επιτυγχάνεται με τις εντολές MOV όπως έχουμε δει για επικοινωνία με εξωτερική μνήμη. Έτσι, η κάρτα γραφικών χαρτογραφείται στο ίδιο εύρος διευθύνσεων με την εξωτερική μνήμη.

## 3. Συναρτήσεις

### 3.1 setgraphicsmode

**(C1) Η συνάρτηση setgraphicsmode**

1. Κατασκευάστε τη συνάρτηση αυτή η οποία θα ρυθμίζει την οθόνη να χρησιμοποιήσει γραφικά (*al=13h, int 10h*).
2. Τοποθετήστε τις αντίστοιχες εντολές PUSH/POP για να είναι διαφανής.
3. Στο κυρίως πρόγραμμα τοποθετήστε την κλήση αυτής της συνάρτησης.

### 3.2 putpixel

**(C2) Η συνάρτηση putpixel**

**(Τοποθέτηση pixel σε ένα σημείο στην οθόνη)**

Η συνάρτηση αυτή θα τοποθετεί ένα pixel με τιμή 9 στις συντεταγμένες που ορίζονται από το ζεύγος  $(x,y)=(BX,AX)$ . Η διεύθυνση που ξεκινάει η μνήμη της κάρτας γραφικών είναι η A000h και αντιστοιχεί στο πάνω αριστερό pixel.

1. Δώστε την τιμή 320 σε ένα διαθέσιμο καταχωρητή 16 bit (όχι στους BX,AX γιατί αυτοί χρησιμοποιούνται).
2. Κάντε τον πολλαπλασιασμό  $AX * 320$ , χρησιμοποιώντας τον προηγούμενο καταχωρητή, προκειμένου να υπολογίσουμε την μετατόπιση γραμμής.
3. Προσθέστε την τιμή του BX
4. Το αποτέλεσμα που βρίσκεται στον AX να μεταφερθεί στον καταχωρητή SI.
5. Να μεταφερθεί στον BX η τιμή A000h
6. Να μεταφερθεί η τιμή από τον BX στον ES
7. Να γράψετε την τιμή 9 στη μετατόπιση [SI] του τμήματος που ξεκινάει από το ES.
8. Τοποθετήστε τις αντίστοιχες εντολές PUSH/POP για να είναι διαφανής.

### 3.3 drawline1

#### (C3) Η συνάρτηση drawline1

#### (Σχεδίαση γραμμής χρησιμοποιώντας την putpixel)

Η συνάρτηση αυτή θα σχεδιάζει μια ευθεία γραμμή από τις συντεταγμένες (150,10) έως (185,10) χρησιμοποιώντας διαδοχικές κλήσεις της putpixel.

1. Τοποθετήστε στον καταχωρητή AX την τιμή 10.
2. Τοποθετήστε στον καταχωρητή BX την τιμή 150.
3. Τοποθετήστε στον καταχωρητή CX την τιμή 35 ( $185-150=35$ ).
4. Δημιουργήστε μια ετικέτα μνήμης drawline1\_loop.
5. Δημιουργήστε ένα βρόχο επανάληψης με LOOP ως την ετικέτα drawline1\_loop .
6. Μέσα στο βρόχο επανάληψης:
  1. καλέστε τη συνάρτηση putpixel
  2. να αυξηθεί η BX κατά 1 με την εντολή inc
7. Τοποθετήστε τις αντίστοιχες εντολές PUSH/POP για να είναι διαφανής.
8. Στο κυρίως πρόγραμμα καλέστε τη συνάρτηση και επιβεβαιώστε την ορθή λειτουργία (θα πρέπει να δείτε να σχεδιάζεται μια ευθεία γραμμή).
9. Απομακρύνετε τις γραμμές που τοποθετούνται τιμές στους AX, BX, CX και τοποθετήστε τις στο καλών πρόγραμμα, πριν την κλήση της συνάρτησης. Με αυτόν τον τρόπο γίνεται παραμετροποιήσιμη η συνάρτηση.

Η συνάρτηση **drawline** έχει το μειονέκτημα ότι απαιτεί πολλαπλές εκτελέσεις της εντολής MUL (μέσα στη συνάρτηση putpixel) κάτι που αυξάνει τον υπολογιστικό φόρτο. Ένας πολύ καλός τρόπος βελτιστοποίησης είναι να υπολογιστεί μια φορά η συντεταγμένη και στη συνέχεια απλώς να προσθέτουμε 1.

### 3.4 drawline\_horizontal\_right

#### (C4) Η συνάρτηση drawline\_horizontal\_right

#### (Σχεδίαση οριζόντιας γραμμής προς τα δεξιά με βελτιστοποίηση)

- Η συνάρτηση αυτή θα σχεδιάζει μια γραμμή μήκους σε pixels CX, ξεκινώντας από τις συντεταγμένες (x,y)=(BX,AX).
- Η συνάρτησή μας θα χρησιμοποιεί ως είσοδο τους εξής καταχωρητές:
  - **AX** καταχωρητής που θα αντιστοιχεί στη συντεταγμένη y.
  - **BX** καταχωρητής που θα αντιστοιχεί στη συντεταγμένη x.
  - **CX** καταχωρητής που θα αντιστοιχεί στο μήκος της γραμμής (δηλαδή στον αριθμό των pixels).

#### Οδηγίες:

1. Αρχικά θα υπολογίσουμε τη θέση του pixel. Όπως προηγουμένως κάντε τον πολλαπλασιασμό  $AX * 320 + BX$ . Προσέξτε τώρα ότι ο καταχωρητής **CX** χρησιμοποιείται, οπότε θα πρέπει να τοποθετήσουμε την τιμή 320 στον καταχωρητή **DX**. Μετά τον πολλαπλασιασμό προσθέστε το BX.
2. Μεταφέρετε το αποτέλεσμα του πολλαπλασιασμού στο SI, όπως προηγουμένως.
3. Τοποθετήστε στο ES την τιμή 0A000h όπως προηγουμένως.
4. Δημιουργήστε μια ετικέτα μνήμης **drawline\_horizontal\_right\_again**:
5. Λίγες γραμμές παρακάτω τοποθετήστε την εντολή loop που να πηγαίνει σε αυτή την ετικέτα.
6. Μέσα στο βρόχο επανάληψης:
  1. τοποθετήστε το pixel τιμής 9 στη θέση μνήμης es:[si]
  2. αυξήστε το si κατά 1 με την εντολή inc
7. Τοποθετήστε τις αντίστοιχες εντολές PUSH/POP για να είναι διαφανής.
8. Στο κυρίως πρόγραμμα τοποθετήστε τις τιμές
  - AX=100 (δηλαδή γραμμή 100)
  - BX=160 (δηλαδή στήλη 160)
  - CX=40 (δηλαδή μήκος 40 pixels)
  - και καλέστε τη συνάρτηση. Επιβεβαιώστε την ορθή λειτουργία (θα πρέπει να δείτε να σχεδιάζεται μια ευθεία γραμμή στις αντίστοιχες συντεταγμένες με μήκος 30 pixels).

### 3.5 drawline\_horizontal\_left

#### (C5) Η συνάρτηση drawline\_horizontal\_left

(Σχεδίαση με βελτιστοποίηση)

- Η συνάρτηση αυτή θα είναι ακριβώς ίδια με τη συνάρτηση drawline\_horizontal\_right μόνο που αντί για inc SI θα έχει την εντολή μείωσης του SI κατά 1 (*dec SI*).
- Στο κυρίως πρόγραμμα τοποθετήστε κάτω από την κλήση της συνάρτησης drawline\_horizontal\_right την κλήση αυτής της συνάρτησης. Θα πρέπει να σχεδιαστεί μια ευθεία 40 μήκους προς τα αριστερά.

### 3.6 drawline\_vertical\_top

#### (C6) Η συνάρτηση drawline\_vertical\_top

(Σχεδίαση κατακόρυφης γραμμής προς την κορυφή με βελτιστοποίηση).

- Η συνάρτηση αυτή θα είναι ακριβώς ίδια με τη συνάρτηση drawline\_horizontal\_right μόνο που αντί για inc SI θα έχει την εντολή μείωσης του SI κατά μια ολόκληρη γραμμή, δηλαδή κατά 320 (*χρησιμοποιήστε την εντολή sub*).
- Στο κυρίως πρόγραμμα τοποθετήστε κάτω από την κλήση της συνάρτησης drawline\_horizontal\_right την κλήση αυτής της συνάρτησης. Θα πρέπει να σχεδιαστεί μια ευθεία μήκους 40 προς την κορυφή.

### 3.7 drawline\_vertical\_bottom

#### (C7) Η συνάρτηση drawline\_vertical\_bottom

(Σχεδίαση προς την κορυφή με βελτιστοποίηση).

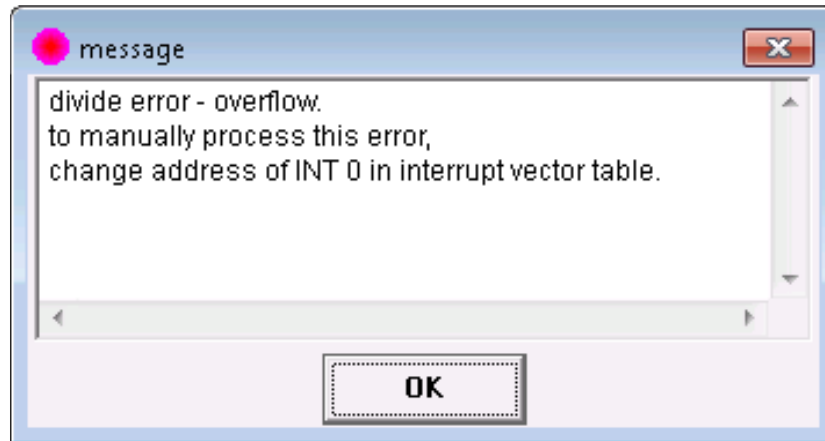
- Η συνάρτηση αυτή θα είναι ακριβώς ίδια με τη συνάρτηση drawline\_horizontal\_right μόνο που αντί για inc SI θα έχει την εντολή αύξησης του SI κατά μια ολόκληρη γραμμή, δηλαδή κατά 320 (*χρησιμοποιήστε την εντολή add*).
- Στο κυρίως πρόγραμμα τοποθετήστε κάτω από την κλήση της συνάρτησης drawline\_vertical\_top την κλήση αυτής της συνάρτησης. Θα πρέπει να σχεδιαστεί μια ευθεία μήκους 40 προς τα κάτω.

### 3.8 Συνάρτηση χειρισμού εξαιρέσεων

Κατά τη λειτουργία του επεξεργαστή δημιουργούνται καταστάσεις που αναγκάζουν τον επεξεργαστή να σταματήσει την εκτέλεση του προγράμματος και να εκτελέσει ειδικό κώδικα (που ονομάζεται κώδικας χειρισμού διακοπής). Τέτοιες καταστάσεις είναι για παράδειγμα η υπερχειλίση ή η διαίρεση με το 0. Επίσης, διακοπές δημιουργούνται από περιφερειακά, ως μια αποτελεσματική τεχνική εισόδου/εξόδου, που είναι πιο αποδοτική από τη επαναληπτική σταθμοσκόπηση (polling).

Σε αυτή την άσκηση θα δούμε πως μπορούμε να χρησιμοποιήσουμε τη δικιά μας συνάρτηση χειρισμού διακοπής.

1. **(A1)** Δημιουργήστε ένα πρόγραμμα το οποίο κάνει μια διαίρεση με το 0. Θα δείτε ότι θα εμφανίζεται η παρακάτω οθόνη, δηλαδή ότι έχει συμβεί η διακοπή (*INT 0*):



2. **(A2)** Βρείτε στη βιβλιογραφία/internet πότε δημιουργείται το INT 1 και πότε το INT 5 (*google: 8086 interrupt vector assignment*).
3. Για να δημιουργήσουμε μια συνάρτηση χειρισμού θα πρέπει να ακολουθήσετε τα παρακάτω βήματα:

- ενεργοποιείτε τα interrupt στο κυρίως πρόγραμμα με την εντολή **STI**
- **(C8)** Δημιουργείτε μια συνάρτηση χειρισμού των interrupt με το όνομα **int0handler**. Μέσα στη συνάρτηση:

1. θα απενεργοποιείτε τα interrupt με την εντολή **CLI**
2. θα αποθηκεύετε όλους τους καταχωρητές που θα χρησιμοποιήσετε με PUSH στην αρχή και θα τους επαναφέρετε με pop πριν το RET
3. θα εκτυπώνετε ένα μήνυμα που θα έχετε τοποθετήσει στο data segment με την 09 κλήση του int 21h (το μήνυμα θα είναι "sorry, division by 0 (AM: XXXX)"), (με AM τον αριθμό μητρώου σας).
4. θα επιστρέφετε στο σημείο που έγινε η κλήση με την **IRET** (επιστροφή από συνάρτηση χειρισμού διακοπής, όχι ret). Επειδή αυτή η συνάρτηση θα εκτελείται σε χρονική στιγμή που δεν ξέρουμε την τιμή του καταχωρητή DS, θα πρέπει πριν από την εκτύπωση να τοποθετήσετε στο DS την τιμή που δείχνει που ξεκινάει το τμήμα δεδομένων, όπως ακριβώς οι 2 γραμμές του κυρίως προγράμματος στο template (A) ή στο template (B).

**(A3)** Ποιος είναι ο machine code για την εντολή STI και ποιος για τη CLI, και πως το βρήκατε;



- **(C9)** Δημιουργείτε μια συνάρτηση αρχικοποίησης του διανύσματος χειρισμού διακοπής με όνομα `int0init` ως εξής:
  1. Θα είναι διαφανής.
  2. Θα χρησιμοποιήσουμε το `int 21h` για να θέσουμε τη νέα διεύθυνση χρησιμοποιώντας την 25h κλήση του DOS ως εξής:
    - **AH=25h**,
    - στον AL τον αριθμό του διανύσματος (στη συγκεκριμένη περίπτωση μας είναι 0, όπως βρήκαμε προηγουμένως),
    - στο **DS:DX** τη διεύθυνση της συνάρτησης χειρισμού της διακοπής. Για αυτό το λόγο μεταφέρετε στο DS τη διεύθυνση του τμήματος που είναι η συνάρτηση `int0handler` η οποία είναι η τιμή του CS, ενώ στο DX μεταφέρετε την μετατόπιση της συνάρτησης μέσα στο τμήμα όπως υπολογίζεται με το **LEA DX,int0handler** .
  3. Επιστρέψτε στο κυρίως πρόγραμμα με RET αφού είναι μια απλή συνάρτηση.
- Στο κυρίως πρόγραμμα καλέστε τη συνάρτηση `int0init` για να εγκαταστήσετε τη διακοπή.
- Στη συνέχεια ακολουθεί η διαίρεση με το 0

Κατά την εκτέλεση θα πρέπει να δείτε να εμφανίζεται το string που δώσατε, αντί για το μήνυμα του προσομοιωτή. Αν γίνει αυτό τότε συγχαρητήρια έχετε εγκαταστήσει με επιτυχία την πρώτη συνάρτηση χειρισμού εξαιρέσεων.

**(A4)** Δώστε ένα screenshot στο οποίο εμφανίζεται το παράθυρο με το μήνυμά σας. Να αποθηκευτεί σε ξεχωριστό αρχείο (*a4.png*).

### 3.9 Συνάρτηση χειρισμού διακοπής για περιφερειακό

Με παρόμοιο τρόπο με το παραπάνω, θα δημιουργήσετε μια συνάρτηση, η οποία θα εξυπηρετεί ένα εξωτερικό περιφερειακό που είναι συνδεδεμένο στον επεξεργαστή. Προκειμένου να εξομοιωθεί το interrupt από μια περιφερειακή συσκευή ο emu8086 διαβάζει συνεχώς το αρχείο <c:\emu8086.hw> του οποίου κάθε byte αντιστοιχεί σε μια συσκευή (και σε μια διακοπή κατ' επέκταση). Αν γράψετε π.χ. μια μη μηδενική τιμή στο πρώτο byte τότε θα ενεργοποιηθεί αυτόματα το interrupt 0, και η τιμή αυτή θα γίνει 0 μόλις αρχίσει η εκτέλεση της συνάρτησης χειρισμού διακοπής. Αν δεν υπάρχει το αρχείο το δημιουργείτε εσείς με το notepad (όχι με εντολές assembly).

Επιλέξτε έναν αριθμό διακοπής συσκευής (π.χ. το 1), δημιουργήστε κατάλληλο κώδικα, ώστε μόλις εμφανιστεί η διακοπή που επιλέξατε να εμφανίζει ένα μήνυμα **"interrupt at device 1 detected (AM:XXXX)"**, και δημιουργήστε ένα πρόγραμμα το οποίο έχει κάποιες εντολές σε ένα αέναο βρόχο (δηλαδή έχει πριν το τέλος, μια εντολή **JMP START**).

Ανοίξτε με κατάλληλο πρόγραμμα (τελευταία σελίδα) το αρχείο με τα hardware interrupts, γράψτε στην κατάλληλη θέση μια τιμή (π.χ. στη θέση 1 για το interrupt 1), θα δείτε ότι θα εκτυπωθεί στην οθόνη το συγκεκριμένο μήνυμα. Αν γίνει αυτό τότε συγχαρητήρια έχετε εγκαταστήσει με επιτυχία την πρώτη συνάρτηση χειρισμού διακοπής εξωτερικής συσκευής.

**Προσοχή:** Αν χρησιμοποιείτε το emu8086.virtualpackage.exe τότε η διαδρομή των hardware interrupts δεν είναι η c:\emu8086.hw αλλά η διαδρομή που αντιστοιχεί στο εικονικό περιβάλλον (φάκελος VOS, υποκατάλογος της εφαρμογής emu8086).

Ο φάκελος **VOS** (virtual operating system) μπορεί να βρεθεί αν ανοίξετε ένα τερματικό windows (**Έναρξη** → **Εκτέλεση** → **cmd**) και στο μαύρο παράθυρο δώστε:

```
cd c:\
```

```
dir vos /s /a
```

με τις παραπάνω εντολές θα σας εμφανιστεί που βρίσκεται ο φάκελος VOS π.χ. **Directory of C:\Users\user\AppData\Roaming**. Μέσα στο φάκελο VOS θα βρείτε το φάκελο για την εφαρμογή μας (π.χ. emu8086 microprocessor emulator), και μέσα σε αυτό το φάκελο, θα βρείτε το φάκελο C\_\ ο οποίος είναι ο εικονικός τοπικός δίσκος για τη εφαρμογή μας. Οπότε, στο παράδειγμά μας, η πλήρη διεύθυνση για τα interrupt είναι η:

```
C:\Users\user\AppData\Roaming\VOS\emu8086 microprocessor emulator\C_\
```

## 4. Ερωτήσεις κατανόησης

1. **(A5)** Γιατί μέσα στη συνάρτηση χειρισμού διακοπής δίνουμε **CLI**;
2. **(A6)** Τι θα συμβεί στον κώδικά μας, αν κατά τη διάρκεια χειρισμού διακοπής συμβεί μια επιπρόσθετη διακοπή;
3. **(A7)** Οι διευθύνσεις των χειριστών διακοπών αποθηκεύονται στη RAM. Σε ποια διεύθυνση βρίσκεται ο χειριστής διακοπής για το **int 0** (χρησιμοποιήστε βιβλιογραφία/internet<sup>1</sup>); Επιβεβαιώστε τη διεύθυνση που θα βρείτε με το να ανοίξετε το memory window (**view** → **memory** στον emulator) και παρατηρήστε το κατά την βηματική εκτέλεση, πριν και μετά την εκτέλεση της συνάρτησης αρχικοποίησης **int0init**.

---

<sup>1</sup> google: interrupt vector table ή σχετική καταχώρηση στη βοήθεια, στο τέλος του φυλλαδίου

## 5. Βοήθεια με Ερωτήσεις και Απαντήσεις

- Τι εντολές θα βάλω μέσα στον αέριο βρόχο;

Βάλτε εντολές για τον υπολογισμό μιας μαθηματικής συνάρτησης, π.χ.  $x^3$

- Το αρχείο `emu8086.hw` τι μορφή πρέπει να έχει;

Είναι ένα απλό αρχείο. Αρχικά θα πρέπει να είναι τελείως άδειο (χωρίς κανένα χαρακτήρα, ούτε καν space)

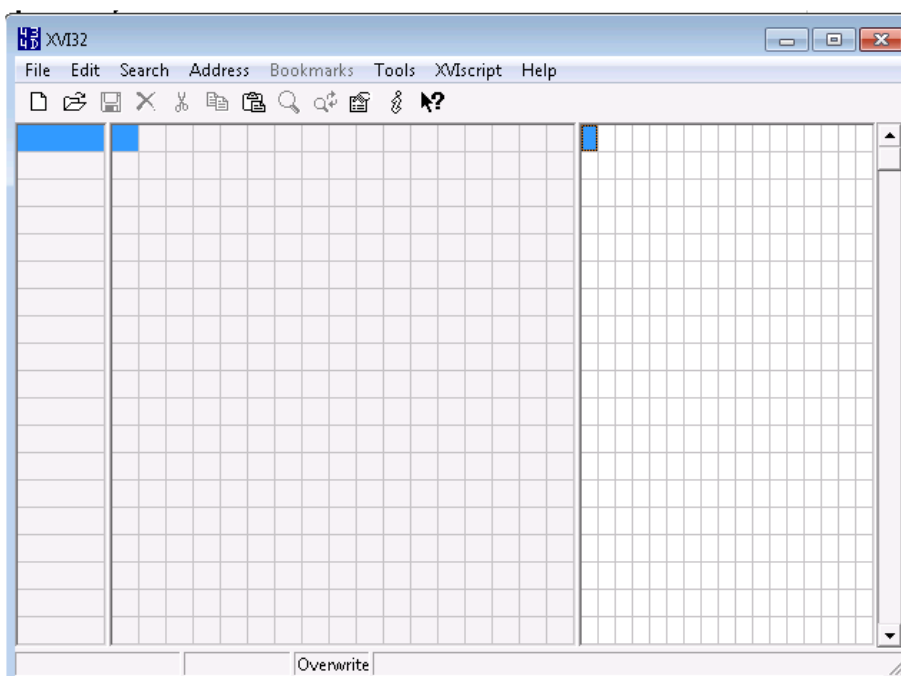
- Το αρχείο `emu8086.hw` το δημιουργώ εγώ, και αν ναι που;

Αν δεν υπάρχει πρέπει να το δημιουργήσετε εσείς στον κεντρικό κατάλογο (ρίζα) <c:\lemu8086.hw> όχι μέσα σε υποκατάλογο.

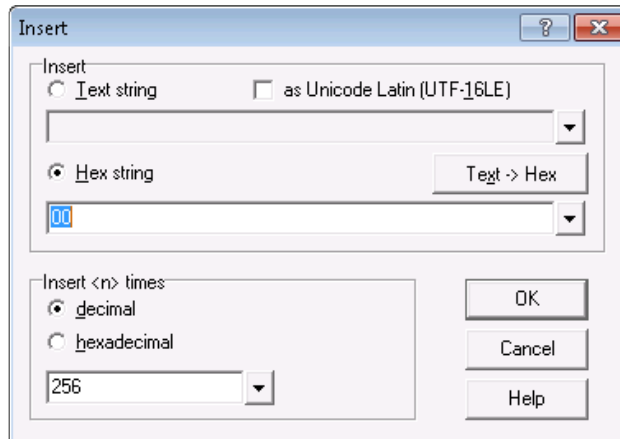
(δείτε τη σημείωση παραπάνω για το αν χρησιμοποιείτε το εικονικό πακέτο `emu8086.exe`)

- Πως θα γράψω ένα χαρακτήρα σε μια θέση στο `emu8086.hw`;

Δε μπορείτε να χρησιμοποιήσετε το word ή το notepad. Θα πρέπει να χρησιμοποιήσετε ένα πρό-γραμμα δεκαεξαδικής επεξεργασίας όπως το hexedit, `xvi32`.



Ανοίγετε το αρχείο, πατάτε το πρώτο κουτάκι και στη συνέχεια από το μενού EDIT→ Insert String δίνετε (Hex String 00 και Εισαγωγή 256 φορές).



- Πως επιβεβαιώνεται ότι λειτούργησε σωστά το interrupt;

Καθώς εκτελείται το πρόγραμμα στο emulator τοποθετήστε στο κουτάκι που θέλετε να χειριστείτε το interrupt την τιμή 01 και θα δείτε ότι θα ενεργοποιηθεί στο emulator το κατάλληλο interrupt. Θα δείτε στο παράθυρο του emulator το μήνυμα “Hardware Interrupt: XX”



- Τι θα συμβεί μετά την εμφάνιση του μηνύματος Hardware Interrupt;

Ο emu8086 θα απομακρύνει την τιμή που είχατε τοποθετήσει στο **emu8086.hw**. Αν κλείσετε και ανοίξετε πάλι το αρχείο θα δείτε ότι στη θέση που είχατε τοποθετήσει το 1, θα υπάρχει 0.

- Δομή και διευθύνσεις των συναρτήσεων χειρισμού διακοπών στο 8086/88.

