



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

Αρχιτεκτονική Υπολογιστών

Ασκήσεις Εργαστηρίου

Ενότητα: ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ Νο 06

Δρ. Μηνάς Δασυγένης

mdasyg@ieee.org

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

[http:// arch.ict.e.uowm.gr/mdasyg](http://arch.ict.e.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.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Περιεχόμενα

1. Σκοπός Άσκησης	4
2. Εργαστηριακή Άσκηση.....	4
2.1 Ενδεικτικές Οδηγίες- Ερωτήσεις	5
3. Ερωτήσεις Κατανόησης	8

1. Σκοπός Άσκησης

- Διευθυνσιοδότηση μνήμης.
- Πίνακες σε assembly.

(A) 25 Ερωτήσεις

(C) 1 Πρόγραμμα

2. Εργαστηριακή Άσκηση

ΠΡΟΓΡΑΜΜΑ C1

Να συντάξετε πρόγραμμα με το οποίο έπειτα από μήνυμα προς τον χρήστη, θα εισάγεται με ταυτόχρονη εμφάνιση στην οθόνη από το πληκτρολόγιο κείμενο με **λατινικούς χαρακτήρες** (ύστερα από έλεγχο) και θα αποθηκεύονται σε μια περιοχή μνήμης που έχει οριστεί στο τμήμα δεδομένων μόνο οι κεφαλαίοι (**A - Z**) και οι μικροί (**a-z**) λατινικοί χαρακτήρες και τα σημεία στίξης: κενό (**space**), θαυμαστικό (**!**) και τελεία (**.**).

Η περιοχή αυτή θα είναι δηλωμένη με το όνομα **buffer** και θα είναι μεγέθους 40 bytes. Για να δηλώσουμε μια θέση μνήμης στο τμήμα δεδομένων με κάποιον αριθμό από Bytes θα χρησιμοποιήσουμε έναν από τους παρακάτω τρόπους. Για παράδειγμα αν θέλουμε μια μεταβλητή πίνακα με όνομα **array** και μέγεθος 3 bytes το κάνουμε με 2 τρόπους:

- (**α**) είτε να δώσουμε 3 φορές τιμή χωρισμένη με κόμμα:

```
array1 db 0,0,0
```

- (**β**) είτε να χρησιμοποιήσουμε το 3 dup (0) που σημαίνει επανέλαβε 3 φορές το byte που βρίσκεται μέσα στην παρένθεση δηλαδή το 0:

```
array1 db 3 dup(0)
```

Η εισαγωγή θα τερματίζεται:

- αν πατηθεί το Enter ή
- αν έχουμε εισάγει 40 χαρακτήρες.

Το πρόγραμμα θα ελέγχει εάν έχουν εισαχθεί κάποιοι χαρακτήρες, και στην περίπτωση που δεν έχουμε εισάγει τίποτα θα εμφανίζει κατάλληλο μήνυμα και θα τερματίζεται.

Αν όμως έχουμε εισάγει κάποιους χαρακτήρες, θα εμφανίζει στην οθόνη μήνυμα ακολουθούμενο από όσους χαρακτήρες έχουμε αποθηκεύσει στο buffer, αφού **μετατραπούν** τα κεφαλαία σε μικρά ενώ τα μικρά σε κεφαλαία (δε θα γίνει άλλη τροποποίηση).

Στο τέλος θα εκτυπώνεται και ο αριθμός των έγκυρων χαρακτήρων. Τα σημεία στίξης θα παραμένουν ίδια. Π.χ.

Eisagete keimeno: This is a text. A SMALL one!

To keimeno meta tin metatropi: tHIS IS A TEXT. a small ONE!

28

2.1 Ενδεικτικές Οδηγίες-Ερωτήσεις

(μπορείτε να χρησιμοποιήσετε διαφορετική υλοποίηση)

Βρείτε τους ASCII χαρακτήρες που θα χρησιμοποιήσουμε στο πρόγραμμα:

(A1) Space

(A2) Θαυμαστικό

(A3) Τελεία

(A4) Enter

(A5) μικρό 'a'

(A6) μικρό 'z'

(A7) κεφαλαίο 'A'

(A8) κεφαλαίο 'Z'

Ορίστε στο τμήμα δεδομένων:

- ένα μήνυμα προτροπής με επικέτα **promptmsg** “Eisagete keimeno:”
- ένα μήνυμα μετατροπής με επικέτα **metatrmsg** “To keimeno meta tin metatropi:”
- ένα μήνυμα ότι δεν έδωσε χαρακτήρες με επικέτα **nochars** “Den dwsate xarakthres”
- μια μεταβλητή buffer με 40 bytes αρχικής τιμής 0

αποφασίζουμε να χρησιμοποιήσουμε έναν καταχωρητή για να μετράμε τους έγκυρους χαρακτήρες που δίνει ο χρήστης. Κάθε φορά που ο χρήστης δίνει έναν έγκυρο χαρακτήρα θα αυξάνεται κατά 1 αυτός ο μετρητής. Έστω επιλέγουμε τον καταχωρητή **SI** για αυτήν τη λειτουργία (συνήθως επιλέγεται το **SI** ή **DI** για μετρητές). Η αύξηση του καταχωρητή κατά 1 θα γίνεται με την εντολή **inc** ως εξής:

inc SI

ΣΗΜΕΙΩΣΗ: Η επιλογή του καταχωρητή **SI** **ΔΕΝ** είναι τυχαία. Ο καταχωρητής αυτός είναι καταχωρητής διευθυνσιοδότησης και θα χρησιμοποιηθεί ταυτόχρονα για τη διευθυνσιοδότηση μνήμης με μετατόπιση (displacement), και συγκεκριμένα για την αποθήκευση των χαρακτήρων.

1. Ξεκινήστε με το **template (A)**
2. Θα πρέπει να κατασκευάσουμε ένα βρόχο επανάληψης από 0 έως 40, αυτό το κάνουμε γιατί το μέγιστο πλήθος χαρακτήρων που θέλουμε είναι 40.
 1. Μηδενίζουμε το counter μας ως εξής: **MOV SI,0**
 2. Από κάτω τοποθετούμε την επικέτα **startloop1**: γιατί σε αυτό το σημείο θα επιστρέφουμε με ένα **jmp** για την επόμενη επανάληψη.
 3. Αρκετές γραμμές παρακάτω τοποθετούμε τον έλεγχο ότι εφόσον είμαστε κάτω από 40 χαρακτήρες να επιστρέφουμε στην αρχή του loop. Επίσης, κάτω από τις εντολές αυτές θα έχουμε μια επικέτα **endloop1**: η οποία θα χρησιμοποιείται αν θέλουμε να μεταβούμε σε αυτό το σημείο αν υπάρξει λόγος (δηλαδή αν ο χρήστης πατήσει ENTER) ως εξής

```

cmp SI,40
JB startloop1
endloop1:

```

4. Τοποθετήστε επίσης στη ανάμεσα από αυτές τις γραμμές μια ακόμη επικέτα μνήμης **storeloop1**: η οποία θα χρησιμοποιείται για να μεταβούμε από κάποιον έλεγχο αν θέλουμε να αποθηκεύσουμε τον χαρακτήρα.
5. Εμφανίστε το μήνυμα προτροπής **promptmsg**
6. Μέσα στο βρόχο επανάληψης θα κάνετε τα εξής βήματα πριν την επικέτα **storeloop1**:
 1. Εισαγωγή χωρίς εμφάνιση ενός χαρακτήρα.
 2. AN ο χαρακτήρας είναι το **enter** τότε πήγαινε στο **endloop1**
 3. AN ο χαρακτήρας είναι το **!** Πήγαινε στο **storeloop1**:
 4. AN ο χαρακτήρας είναι το **space** Πήγαινε στο **storeloop1**:
 5. AN ο χαρακτήρας είναι κάτω από την ASCII τιμή του 'A' τότε σημαίνει ότι είναι μη αποδεκτός χαρακτήρας (**A9**) (γιατί ισχύει αυτό;) και πήγαινε στην αρχή του loop **startloop1**:
 6. AN ο χαρακτήρας είναι κάτω από 'Z' τότε σημαίνει ότι είναι αποδεκτός **ΚΕΦΑΛΑΙΟΣ** χαρακτήρας (**A10**) (Πως είμαστε σίγουροι ότι ισχύει αυτό;) και πήγαινε στο **storeloop1**:
 7. AN ο χαρακτήρας είναι κάτω από 'a' τότε σημαίνει ότι είναι μη αποδεκτός χαρακτήρας (**A11**) (γιατί ισχύει αυτό;) και πήγαινε στην αρχή του loop **startloop1**:
 8. AN ο χαρακτήρας είναι κάτω από 'z' τότε σημαίνει ότι είναι αποδεκτός **μικρός** χαρακτήρας (**A12**) (Πως είμαστε σίγουροι ότι ισχύει αυτό;) και πήγαινε στο **storeloop1**:
 9. Τοποθετήστε ένα **jmp** (διακλάδωση χωρίς συνθήκη) για να μεταβείτε στο **startloop1**: γιατί όλοι οι άλλοι χαρακτήρες είναι άκυροι.
7. Μέσα στο βρόχο επανάληψης και αμέσως μετά την επικέτα **storeloop1**:

1. Θα εμφανίσετε το χαρακτήρα που μόλις έχει εισαχθεί. Σε αυτό το σημείο έχουμε σίγουρα έγκυρο χαρακτήρα. Ο χαρακτήρας είναι σε μορφή ASCII και δε χρειάζεται μετατροπή **(A13)** *(Γιατί ο χαρακτήρας είναι ήδη σε μορφή ASCII;)*
2. Θα τοποθετήσετε τον χαρακτήρα στη θέση μνήμης `buffer[si]` ως εξής:
MOV buffer[si] , al
3. Αμέσως μετά θα αυξήσετε το SI κατά 1, ως εξής **inc si**.
8. Δε χρειάζεται κάτι άλλο σε αυτό το βρόχο.
9. Αμέσως μετά την ετικέτα **endloop1** κάντε έναν έλεγχο της **SI**. Αν η **SI** έχει τιμή 0, δηλαδή δεν έχουμε εισάγει κάποιον χαρακτήρα τότε θα πρέπει να εμφανίζεται το μήνυμα **nochars** και να πηγαίνουμε στο τέλος του προγράμματος.
10. Εκτυπώστε το μήνυμα **metatrmmsg**.
11. Τοποθετήστε την ετικέτα **startloop2**:
12. Σε αυτό το βρόχο θα χρησιμοποιήσουμε τη μεταβλητή DI για να μετράμε τις επαναλήψεις. Ο αριθμός των επαναλήψεων θα είναι από 0 έως SI **(A14)** *(Γιατί από 0 ως SI και όχι από 0 έως 40;)*.
13. Αρχικά διαβάστε σε έναν καταχωρητή τον αντίστοιχο χαρακτήρα από τον πίνακα `buffer` ως εξής: **MOV AL, buffer [DI]**
14. Αυξήστε την τιμή του DI κατά 1
15. Αρκετές γραμμές παρακάτω δώστε τις γραμμές που σηματοδοτούν το τέλος του βρόχου, με παρόμοιο τρόπο όπως στο προηγούμενο βρόχο.
16. Βρείτε την απόσταση που έχουν οι μικροί και οι κεφαλαίοι. Δηλαδή, βρείτε την ASCII τιμή του 'a' και την ASCII τιμή του 'A' **(A15)** *(ποια είναι η σταθερή απόσταση που έχουν οι μικροί και οι κεφαλαίοι χαρακτήρες;)*.
17. Θα πρέπει να κάνετε τους εξής ελέγχους. Αν είναι οι χαρακτήρες θαυμαστικό, κενό, τελεία δε θα γίνεται καμία μετατροπή και θα πηγαίνει η εκτέλεση κάποιες γραμμές παρακάτω σε μια ετικέτα **printchar**:
18. Αν είναι από **a-z** θα μετατρέπεται στον αντίστοιχο κεφαλαίο προσθέτοντας ή αφαιρώντας τη σταθερή τιμή που υπολογίσατε πιο πριν και θα πηγαίνει η εκτέλεση σε μια ετικέτα **printchar**:
19. Αν είναι από **A-Z** θα μετατρέπεται στον αντίστοιχο μικρό προσθέτοντας ή αφαιρώντας τη σταθερή τιμή που υπολογίσατε πιο πριν και θα πηγαίνει η εκτέλεση σε μια ετικέτα **printchar**:
20. Μετά την ετικέτα **printchar**: Εμφανίστε το χαρακτήρα.
21. Ακολουθεί ο έλεγχος αν ο δείκτης SI έχει γίνει ίσος ή μικρότερος με το DI και αντιστοίχως πηγαίνουμε στην αρχή του βρόχου επανάληψης 2 ή τερματίζουμε το βρόχο.
22. Μετά το τέλος του βρόχου:
Ο αριθμός των έγκυρων χαρακτήρων είναι μέσα στη μεταβλητή SI (2 Byte) Κάντε τις ανάλογες ενέργειες ώστε να εκτυπώσετε τα 2 ψηφία του SI στο δεκαδικό σύστημα.

3. Ερωτήσεις Κατανόησης

16. Αν αντί για το DI χρησιμοποιήσω τον CX με τον ίδιο τρόπο θα είναι σωστό ή λάθος και γιατί; Δηλαδή **MOV AL, BUFFER[CX]**. (A16)
17. Εκτός από **SI** και **DI** ποιοι άλλοι καταχωρητές θα μπορούσαν να χρησιμοποιηθούν με τον ίδιο ακριβώς τρόπο για τη διευθυνσιοδότηση του buffer και γιατί; (κάποιοι καταχωρητές δε μπορούν να χρησιμοποιηθούν για αυτό το σκοπό). (A17)
18. Ποια ισοδύναμη εντολή θα μπορούσε να χρησιμοποιηθεί με την **inc SI** (A18)
19. Τροποποιήστε το πρόγραμμά σας, ώστε ο δεύτερος βρόχος να χρησιμοποιεί την εντολή LOOP για τις επαναλήψεις από 0 έως SI. Δώστε μέσα στο κείμενο των απαντήσεων αυτό το κομμάτι κώδικα μόνο. (A19)
20. Η παρακάτω εντολή έχει λάθος; Και αν ναι είναι λογικό ή συντακτικό; (A20)
cmp DI,SI
JMP startloop1
JA endloop1
21. Η παρακάτω εντολή έχει λάθος; Και αν ναι είναι λογικό ή συντακτικό; **MOV DL,SI**. (A21)
22. Η παρακάτω εντολή έχει λάθος; Και αν ναι είναι λογικό ή συντακτικό; **MUL AL,DL**. (A22)
23. Αν στο τέλος του **promptmsg** δε βάλετε **\$** για να τερματίσει τι θα γίνει; Δοκιμάστε το και δικαιολογήστε τι έγινε. (A23)
24. Πως θα μπορούσαμε να κάνουμε τους ελέγχους αν θέλαμε να ελέγχουμε πρώτα τα κεφαλαία μετά τα μικρά και στο τέλος τους ειδικούς χαρακτήρες; Δώστε μόνο το κομμάτι του κώδικα του πρώτου τμήματος του βρόχου στο οποίο γίνονται οι έλεγχοι. (A24)
25. Τροποποιήστε το πρόγραμμα ώστε να δέχεται στην είσοδο και τους αριθμούς 0-9. Οι αριθμοί αυτοί στην εκτύπωση στη συνέχεια θα μένουν αμετάβλητοι. Δώστε το καινούργιο πρόγραμμα. (A25)