ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

Αρχιτεκτονική Υπολογιστών

Ασκήσεις Εργαστηρίου

Ενότητα: ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ No 07

Δρ. Μηνάς Δασυγένης

mdasyg@ieee.org

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

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. Ερωτήσεις κατανόησης..... 7

1. Σκοπός της άσκησης

- Κλήσεις και επιστροφή συναρτήσεων

(A) 9 Ερωτήσεις
(C) 8 Προγράμματα

2. Εργαστηριακή Άσκηση

Να γραφεί πρόγραμμα με υποπρογράμματα. Στο κυρίως πρόγραμμα θα εισάγετε εσείς κείμενο έπειτα από μήνυμα προτροπής. Η συμβολοσειρά θα είναι το πολύ 80 χαρακτήρων και μπορούμε να σταματήσουμε την εισαγωγή με το χαρακτήρα **'#'**. Ένα υποπρόγραμμα θα διαβάζει από το πληκτρολόγιο ένα χαρακτήρα έπειτα από κατάλληλο μήνυμα. Στη συνέχεια θα εμφανίζει πόσες φορές βρήκε αυτό το χαρακτήρα στη συμβολοσειρά, ενώ αν δεν υπάρχει θα εμφανίζει κατάλληλο μήνυμα. Αν το πλήθος των εμφανίσεων είναι διψήφιος αριθμός, θα εμφανίζεται με τη μορφή δεκάδα-μονάδα. Ξεκινήστε από το template (B) όπου υπάρχει η δήλωση τμήματος σωρού. Επειδή θα χρησιμοποιήσουμε υποσυναρτήσεις απαιτείται να ορίζεται ένα τμήμα σωρού για να τοποθετείται η δ/νση επιστροφής της συνάρτησης.

(C1) Η συνάρτηση printprompt

1. Δημιουργήστε μια συνάρτηση *(PROC)* με το όνομα **printprompt**. Η συνάρτηση αυτή το μόνο που θα κάνει θα είναι να εκτυπώνει το μήνυμα `"Parakalw eisagete keimeno, pathste # gia termatismo"`
2. Στο κυρίως πρόγραμμα τοποθετήστε την κλήση της παραπάνω συνάρτησης (**call printprompt**).
3. Μεταφράστε το πρόγραμμα και εκτελέστε το. Θα πρέπει να εκτυπώνεται το μήνυμα κανονικά.

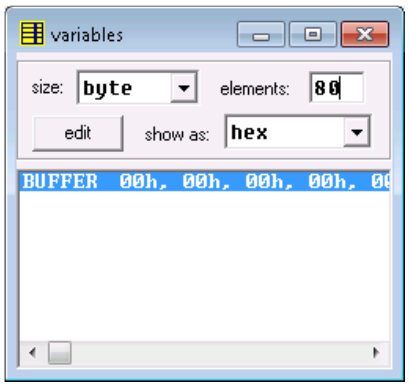
→ Αν το παραπάνω βήμα λειτουργεί ορθά, προχωρήστε στην επόμενη συνάρτηση.

(C2) Η συνάρτηση inputtext

4. Δημιουργήστε μια συνάρτηση *(PROC)* με το όνομα **inputtext**. Η συνάρτηση αυτή θα γεμίζει τον πίνακα **buffer** *(να ορίσετε τον πίνακα buffer στο τμήμα δεδομένων του προγράμματός σας ως db με 80 Bytes)*, με τους χαρακτήρες που βάζει ως χρήστης.
   Βοηθητικές Οδηγίες:
   1. θα μηδενίζει το μετρητή **SI** που θα χρησιμοποιηθεί για τη διευθυνσιοδότηση της μνήμης
   2. θα κάνει είσοδο ενός χαρακτήρα με εμφάνιση.
   3. Θα ελέγχει αν ο χαρακτήρας είναι ο χαρακτήρας τερματισμού (ο **'#'**)
   4. Αν είναι ο χαρακτήρας τερματισμού τότε θα πηγαίνει στο τέλος αυτής της συνάρτησης *(πριν ακριβώς από το RET)*, για να επιστρέφει πίσω.

5. Διαφορετικά, θα τοποθετεί το χαρακτήρα στον πίνακα **buffer** στη θέση **SI** (`mov buffer[si],al`)
6. θα αυξάνει κατά 1 το δείκτη **SI** για να δείξει στην επόμενη θέση.
7. Θα ελέγχει αν ο **SI** είναι ίσος με 80, αφού στην εκφώνηση μας ζητείται να δώσουμε έως 80 χαρακτήρες. Αν δεν είναι ίσος με 80, τότε θα πηγαίνει στο βήμα της εισόδου ενός χαρακτήρα.
8. Θα υπάρχει η εντολή επιστροφής, στο καλών πρόγραμμα RET

5. Στο κυρίως πρόγραμμα τοποθετήστε την κλήση της παραπάνω συνάρτησης κάτω από την προηγούμενη κλήση.
6. Μεταφράστε το πρόγραμμα και εκτελέστε το. Θα πρέπει να μπορείτε να δώσετε χαρακτήρες, και αφού τερματιστεί το πρόγραμμα να πατήστε το κουμπί VARS στο emulator, να επιλέξετε το **buffer**, μέγεθος Byte και 80 στοιχεία ως εξής:



7. Θα πρέπει οι ASCII τιμές που έχετε δώσει να εμφανίζονται στο παράθυρο.

→ Αν το παραπάνω βήμα λειτουργεί ορθά, προχωρήστε στην επόμενη συνάρτηση.

(C3) Η συνάρτηση printinputchar

8. Δημιουργήστε μια συνάρτηση *(PROC)* με το όνομα **printinputchar**. Η συνάρτηση αυτή το μόνο που θα κάνει θα είναι να εκτυπώνει το μήνυμα `"Parakalw dwste to xarakthra poy thelete na brw"`
9. Στο κυρίως πρόγραμμα τοποθετήστε την κλήση της παραπάνω συνάρτησης κάτω από τις προηγούμενες κλήσεις.
10. Μεταφράστε το πρόγραμμα και εκτελέστε το. Θα πρέπει να εκτυπώνεται το μήνυμα κανονικά

→ Αν το παραπάνω βήμα λειτουργεί ορθά, προχωρήστε στην επόμενη συνάρτηση.

(C4) Η συνάρτηση inputchar

11. Δημιουργήστε μια συνάρτηση *(PROC)* με το όνομα **inputchar**. Η συνάρτηση αυτή θα διαβάζει ένα χαρακτήρα με εμφάνιση και θα το τοποθετεί στη θέση μνήμης **chartosearch**, που θα ορίσετε στο τμήμα δεδομένων.
12. Στο κυρίως πρόγραμμα τοποθετήστε την κλήση της παραπάνω συνάρτησης κάτω από τις προηγούμενες κλήσεις.
13. Μεταφράστε το πρόγραμμα και εκτελέστε το. Θα πρέπει να εκτυπώνεται το μήνυμα κανονικά

→ Αν το παραπάνω βήμα λειτουργεί ορθά, προχωρήστε στην επόμενη συνάρτηση.

(C5) Η συνάρτηση searchchar

14. Δημιουργήστε μια συνάρτηση *(PROC)* με το όνομα **searchchar**. Η συνάρτηση αυτή θα ψάχνει μέσα στο **buffer** το χαρακτήρα **chartosearch**. Θα ξεκινάει από **SI=0** έως **SI=80.** Κάθε φορά που βρίσκεται ο χαρακτήρας θα αυξάνεται ένας μετρητής κατά 1.
    Βοηθητικές Οδηγίες:
    1. Μηδενίστε το μετρητή που θα χρησιμοποιήσετε για να μετράει πόσες φορές υπάρχει ο χαρακτήρας στο **buffer**. Αυτό μπορεί να είναι είτε καταχωρητής είτε θέση μνήμης.
    2. Μηδενίστε το **SI** που θα χρησιμοποιήσετε για τη διευθυνσιοδότηση του **buffer[si]**.
    3. Δημιουργήστε μια δομή επανάληψης με **LOOP**, και θέστε στο **CX** την τιμή 80.
    4. Τοποθετήστε το χαρακτήρα **chartosearch** στον καταχωρητή **DL.**
    5. Συγκρίνετε το DL με το **buffer[si].**
    6. Αν είναι ίδιος να αυξάνεται ο μετρητής κατά 1.
    7. Αν είναι διαφορετικός να μη γίνει τίποτα.
    8. Να αυξηθεί κατά 1 το **SI**
    9. Να επαναληφθεί η διαδικασία έως το 5 *(εντολή LOOP).*
15. Στο κυρίως πρόγραμμα τοποθετήστε την κλήση της παραπάνω συνάρτησης κάτω από τις προηγούμενες κλήσεις.
16. Μεταφράστε το πρόγραμμα και εκτελέστε το. Θα πρέπει στο τέλος της εκτέλεσης στο emu8086 να δείτε τον αριθμό των εμφανίσεων στο μετρητή που έχετε χρησιμοποιήσει.

→ Αν το παραπάνω βήμα λειτουργεί ορθά, προχωρήστε στην επόμενη συνάρτηση.

(C6) Η συνάρτηση printdec

17. Δημιουργήστε μια συνάρτηση με το όνομα **printdec**. Η συνάρτηση αυτή θα εκτυπώνει τα δύο ψηφία ενός αριθμού.
    Βοηθητικές Οδηγίες:
    1. Τοποθετήστε τον αριθμό που θέλετε να εκτυπώσετε σε έναν καταχωρητή *(έστω **DL**).*
    2. Μεταφέρετε το **DL** στον **AX** για να κάνετε τη διαίρεση με το 10.
    3. Εκτυπώστε πρώτα τη δεκάδα και στη συνέχεια τη μονάδα όπως το έχετε κάνει σε προηγούμενο εργαστήριο.
18. Στο κυρίως πρόγραμμα τοποθετήστε την κλήση της παραπάνω συνάρτησης κάτω από τις προηγούμενες κλήσεις.
19. Μεταφράστε το πρόγραμμα και εκτελέστε το.

Σημείωση: Στις ερωτήσεις κατανόησης, υπάρχουν τα προγράμματα C7 και C8.

3. Ερωτήσεις κατανόησης

Δώστε δικαιολογημένες απαντήσεις:

1. Οι συναρτήσεις που έχετε κατασκευάσει με τον προηγούμενο τρόπο αν και λειτουργούν σωστά, δεν είναι προγραμματιστικά ορθές *(δεν είναι διαφανείς),* γιατί τροποποιούν τιμές σε κάποιους καταχωρητές και δεν επαναφέρουν τις τιμές αυτές που είχαν πριν από το RET. Θα πρέπει αμέσως κάτω από τη γραμμή PROC να αποθηκεύονται οι τιμές των καταχωρητών που τροποποιούνται και πριν από το RET να επαναφέρονται οι τιμές αυτές. Θα χρησιμοποιήσετε το σωρό για να το κάνετε αυτό. Να τροποποιήσετε τις συναρτήσεις ώστε να είναι διαφανείς. **(A1)**
2. Αφού κάνετε τις συναρτήσεις διαφανείς, να μετρήσετε πόσα Byte χρησιμοποιούνται από το σωρό *(μέγιστη χρήση).* **(A2)**
3. Στη συνάρτηση **searchchar** σε ένα βήμα αναγράφεται “Τοποθετήστε το χαρακτήρα **chartosearch** στον καταχωρητή”. Υπάρχει πρόβλημα αν κάνουμε τη σύγκριση του chartosearch με το **buffer[si]** ως εξής **(A3);**
   `cmp chartosearch, buffer[si]`
4. Στη συνάρτηση **searchchar** σε ένα βήμα αναγράφεται: “Ο μετρητής...μπορεί να είναι είτε καταχωρητής είτε θέση μνήμης”. Είναι προτιμότερο να χρησιμοποιηθεί καταχωρητής για να μετράει τις εμφανίσεις παρά θέση μνήμης. Γιατί; **(A4)**
5. Όπως γνωρίζετε η εντολή ADD επηρεάζει όλες τις σημαίες (**CF** {κρατούμενο}, **PF** {bit άρτιας ισοτιμία}, **AF** {βοηθητική σημαία, υποδεικνύει αν έχει δημιουργηθεί κρατούμενο ή borrow στα 4 LSB, δηλαδή στο μισό καταχωρητή}, **ZF** {bit μηδενικού αποτελέσματος}, **SF** {bit προσήμου}, **OV** {bit υπερχείλισης}). Ψάξτε στη βιβλιογραφία και απαντήστε: Ποια είναι η σημασία της σημαίας IF**; (A5)**
6. Προσπαθήστε να εμφανίσετε τις σημαίες αυτές δίνοντας κατάλληλες τιμές στους καταχωρητές AX, BX στην εντολή ADD AX, BX δικαιολογώντας τις επιλογές σας. Με κάθε ζευγάρι τιμών AX,BX θα ενεργοποιείται μια σημαία

κάθε φορά *(οι σημαίες φαίνονται στον emulator πατώντας το flags)* αφού εκτελεστεί η εντολή ADD *(με single step).* **(A6)**
Για παράδειγμα οι 3 παρακάτω εντολές δε αλλάζουν καμία τιμή στον καταχωρητή σημαιών:

```
MOV AX,01
MOV BX,01
ADD AX,BX
```

7. **(C7)** Να γράψετε εντολές σε assembly, οι οποίες θα τοποθετούν την τιμή 1 στον AX αν το bit No2 *(δηλαδή το τρίτο bit)* του BX είναι 1, διαφορετικά θα τοποθετούν την τιμή 0 στον AX.
8. **(C8)** Να γράψετε εντολές σε assembly, οι οποίες απομακρύνουν τα bit 11 έως και 8 από τον καταχωρητή BX και τα τοποθετούν στον καταχωρητή AX ξεκινώντας από την αρχή. Δηλαδή, το bit 8 θα μπει στο bit 0 του AX, ενώ το bit 11 στο bit 3 του AX.
9. Υποθέτοντας ότι ο καταχωρητής τμήματος DS έχει την τιμή 1234h, ο CS την τιμή 2345h και ο BX την τιμή 63h ποια είναι η φυσική διεύθυνση της μνήμης που θα διαβαστεί για να τοποθετηθεί στον AX στην εντολή
   "MOV AX, BYTE PTR [BX]" (A7)
10. Υποθέτοντας ότι ο καταχωρητής τμήματος DS έχει την τιμή 1234h, ο CS την τιμή 2345h και ο BX την τιμή 63h ποια θα είναι η φυσική διεύθυνση που θα εκτελεστεί μετά την εκτέλεση της εντολής JMP 100h **(A8)**
11. Αν θεωρήσουμε ότι έχουμε μια μνήμη 4MB με λέξεις των 32bit. Πόσα bit θα έπρεπε να χρησιμοποιήσουμε για να τη διευθυνσιοδοτήσουμε αν:
    (α) η μνήμη μας είναι διευθυνσιοδοτημένη κατά λέξεις,
    (β) η μνήμη μας είναι διευθυνσιοδοτημένη κατά Bytes**; (A9)**

.