



ΕΡΓΑΣΤΗΡΙΟ 4

Σκοπός του 4^{ου} εργαστηρίου είναι να η εκμάθηση του τρόπου αποθήκευσης δεδομένων στη μνήμη κάνοντας χρήση δομής δεδομένων και συγκεκριμένα κάνοντας χρήση πίνακα. Τα δεδομένα αποθηκεύονται σε ξεχωριστό τμήμα μνήμης από τις εντολές γι' αυτό τον λόγο η δήλωση του πίνακα, που θα χρησιμοποιήσετε, πρέπει να γίνει στο τμήμα `.data`. Στην συνέχεια δίνεται ο τρόπος δήλωσης του πίνακα και φυσικά η σύνταξη των εντολών `load (lw)` και `store (sw)` τις οποίες πρέπει να χρησιμοποιήσετε για αποθήκευση δεδομένων από την μνήμη σε καταχωρητή και για αποθήκευση δεδομένων από καταχωρητή στη μνήμη αντίστοιχα.

➤ Δήλωση πίνακα (στον κώδικά σας)

```
.data # στο τμήμα .data πρέπει να δηλωθεί ο πίνακας
```



```
array_name: .space X # array_name είναι το όνομα του πίνακά μας  
                # με το .space X δηλώνουμε ότι θέλουμε να κρατήσουμε  
                # χώρο ίσο με X byte για τον πίνακα array_name
```

➤ Εντολές

Εντολή `load (lw)`: Η εντολή αυτή αποθηκεύει δεδομένα σε καταχωρητή, τα οποία έχει πάρει από συγκεκριμένη διεύθυνση της μνήμης.

Σύνταξη: <code>lw Rt, Address(Rs)</code> Σημασία: <code>Rt=Memory[Address+Rs]</code>
--

Προσοχή: όπου `Rt`, `Rs` είναι καταχωρητές και όπου `Address` είναι το όνομα του πίνακα που δώσατε στο τμήμα `.data`.

Από την σημασία της εντολής αυτής γίνεται κατανοητό ότι ουσιαστικά του λέμε, πήγαινε στη μνήμη και από την διεύθυνση `[array_name+Rs]` πάρε τα δεδομένα και αποθήκευσέ τα στον καταχωρητή `Rt`.

Εντολή `store (sw)`: Η εντολή αυτή αποθηκεύει δεδομένα από έναν καταχωρητή σε συγκεκριμένη διεύθυνση στη μνήμη.

Σύνταξη: <code>sw Rt, Address(Rs)</code> Σημασία: <code>Memory[Address+Rs]=Rt</code>
--



Χειρισμός πινάκων σε assembly

Κάθε φορά που θέλετε να γεμίσετε ένα πίνακα και για να επεξεργαστείτε τα δεδομένα του στην συνέχεια, πρέπει μέσω ενός καταχωρητή να αποθηκεύετε τα δεδομένα στην μνήμη (δηλαδή χρήση `sw`), ενώ όταν θέλετε να επεξεργαστείτε τα δεδομένα ενός πίνακα, πρέπει να τα μεταφέρετε-αποθηκεύετε από την μνήμη σε έναν καταχωρητή (χρήση `lw`) και να τα επεξεργάζεστε κάνοντας χρήση του καταχωρητή.

Ο χώρος που καταλαμβάνουν τα δεδομένα, εάν αυτά είναι ακέραιοι (κάθε ακέραιος δεσμεύει 4 bytes από την μνήμη), συνεπώς για να αναφερόμαστε στο επόμενο στοιχείο του πίνακα κάθε φορά πρέπει να αυξάνουμε τον δείκτη του πίνακα (π.χ. $A[i]$: i είναι ο δείκτης) ανά τέσσερα. Έτσι αν $i=0$ τότε $A[0]$ (το 1^ο στοιχείο του πίνακα), αν $i=1$ τότε $A[4]$ (το 4^ο byte δηλαδή το 2^ο στοιχείο του πίνακα) κλπ. Η διαδικασία αυτή ονομάζεται ευθυγράμμιση και γίνεται αν εντάξετε στον κώδικά σας μία εντολή η οποία να κάνει τον δείκτη πολλαπλάσιο του τέσσερα δηλαδή: $i=i*4$.

Ασκήσεις

- Να γράψετε πρόγραμμα σε assembly το οποίο να εισάγει δεδομένα σε πίνακα ακεραίων 6 θέσεων και στην συνέχεια να υπολογίζει και να εμφανίζει το άθροισμα των στοιχείων του πίνακα.
- Να γράψετε πρόγραμμα σε assembly το οποίο να εισάγει δεδομένα σε πίνακα ακεραίων 5 θέσεων και στην συνέχεια να υπολογίζει και να εμφανίζει το μέγιστο στοιχείο του πίνακα.

Υπενθύμιση

Τα προγράμματα που θα παραδίδετε, θα βαθμολογούνται για την ορθότητα τους αλλά και για την σωστή μορφοποίηση (στοίχιση, σχόλια, κλπ.).

Σχετικά με την παράδοση των ασκήσεων

Η παράδοση της εργαστηριακής άσκησης θα πρέπει να γίνει μέχρι την **Τρίτη 19/3** και η εξέτασή της θα πραγματοποιηθεί την **Τετάρτη 20/3** και ώρα αναλόγως του εργαστηρίου στο οποίο είστε εγγεγραμμένοι.



Σε περίπτωση απουσίας του φοιτητή από το εργαστήριο είναι υποχρεωτικό να σταλούν οι ασκήσεις σε ηλεκτρονική μορφή. Για την υποβολή των ασκήσεων, θα δημιουργήσετε ένα φάκελο με όνομα, το όνομα, το επώνυμο και τον ΑΜ σας, πχ. οποτα-ερwhηγο-211xxxx, και θα αντιγράψετε σε αυτόν μόνο τα δύο .s αρχεία. Στην συνέχεια θα συμπιέσετε τον φάκελο (rar ή zip). Τέλος θα ανεβάσετε τον συμπιεσμένο φάκελο στο eclass.