

2ο Εργαστήριο Αρχιτεκτονικής Η/Υ: Αναζήτηση σε πίνακα λέξεων 32-bit

A. Ευθυμίου

Παραδοτέο: Τρίτη 22 Οκτώβρη, 23:59

Με αυτή την εργαστηριακή άσκηση θα γράψετε και θα ελέγξετε ένα απλό πρόγραμμα Risc-V assembly με τον Ripes. Θα πρέπει να έχετε μελετήσει μέχρι και το δεύτερο μάθημα για τη γλώσσα assembly του Risc-V (3η διάλεξη του μαθήματος).

1 Παραλαβή του σκελετού της άσκησης

Ακολουθήστε τον σύνδεσμο που υπάρχει στο ecourse για να δημιουργηθεί το δικό σας αποθετήριο της άσκησης και κλωνοποιήστε το με την εντολή:

```
git clone https://github.com/UoI-CSE-MYY505/lab02-ghUsername.git
```

Προσοχή στη δομή των καταλόγων. Όπως και με την άσκηση 1, θα πρέπει να μοιάζει:

```
MYY505Labs
|- myy505Utils
|- lab01-ghUsername
|- lab02-ghUsername
```

Μεταβείτε στον κατάλογο lab02-ghUsername. Εκεί θα βρείτε το αρχείο lab02.s, με έναν πολύ μικρό σκελετό κώδικα και το Lab02Test.py, που αυτή τη φορά δεν χρειάζεται αλλαγές.

2 Η άσκηση

Θα γράψετε ένα πρόγραμμα που αναζητεί έναν 32-bit αριθμό, που δίνεται στον καταχωρητή a2, σε ένα πίνακα λέξεων 32-bit. Ο πίνακας θα ξεκινά από την διεύθυνση μνήμης που δίνεται στον καταχωρητή a0 και έχει μέγεθος, πλήθος στοιχείων, που δίνεται στον καταχωρητή a1.

Το αποτέλεσμα της αναζήτησης θα είναι η διεύθυνση μνήμης του στοιχείου του πίνακα που περιέχει τον αριθμό. Το αποτέλεσμα θα δίνεται από το πρόγραμμα στον καταχωρητή s0. Αν δεν βρεθεί ο αριθμός στον πίνακα, το αποτέλεσμα θα είναι η τιμή-διεύθυνση 0. Αν υπάρχουν πολλά στοιχεία του πίνακα που ταιριάζουν, το αποτέλεσμα της αναζήτησης θα είναι η διεύθυνση του τελευταίου που ταιριάζει.

Το πρόγραμμά σας θα πρέπει να αποτελεί γενική λύση: να μπορεί να λειτουργήσει σωστά για οποιεσδήποτε τιμές των a0, a1, a2, και για οποιοδήποτε περιεχόμενο του πίνακα, όχι μόνο για το συγκεκριμένο παράδειγμα του αρχείου lab02.s. Για παράδειγμα, ο πίνακας μπορεί να είναι άδειος, οπότε το πλήθος των στοιχείων (a1) θα δίνεται ως 0. Μπορείτε να υποθέσετε ότι η διεύθυνση δεν θα είναι 0 (null), καθώς ο άδειος πίνακας δίνεται από μηδενικό πλήθος στον a1. Επίσης μπορείτε να υποθέσετε ότι το πλήθος (a1) δεν θα είναι ποτέ αρνητικός αριθμός.

Δώστε προσοχή στις λεπτομέρειες! Ζητείται η διεύθυνση μνήμης και όχι η θέση του πίνακα. Επίσης ζητείται η διεύθυνση του **τελευταίου** στοιχείου που ταιριάζει, όχι του πρώτου ή κάποιου ενδιάμεσου.

3 Παραδοτέο και κριτήρια αξιολόγησης

Το παραδοτέο της άσκησης είναι το αρχείο lab02.s που περιέχει το πρόγραμμά σας. Μην αλλάξετε το όνομα του αρχείου, γράψετε εκτός της περιοχής που δείχνουν τα σχόλια, γιατί δεν θα το βρίσκει ο αυτόματος έλεγχος! Προαιρετικά αλλάξτε και το README.md.

Πρέπει να κάνετε commit τις αλλαγές σας και να τις στείλετε (push) στο αποθετήριό σας στο GitHub πριν από την καταληκτική ημερομηνία!

Στις εξετάσεις εργαστηρίου, τα προγράμματά σας θα βαθμολογούνται για την ορθότητά τους, την ποιότητα σχολίων και τη ταχύτητα εκτέλεσής τους. Το τελευταίο σημαίνει ότι πρέπει να είναι σύντομα και ο αριθμός εντολών, ειδικά μέσα σε βρόχο, να είναι όσο γίνεται μικρότερος. Μπορείτε να βρείτε πόσες εντολές εκτελούνται στο tab processor του Ripes (μοιάζει με ολοκληρωμένο κύκλωμα), κάτω δεξιά στο Execution info, πεδίο Instrs. retired.

Στις λύσεις υπάρχουν μερικές παραλλαγές και εκεί μπορείτε να δείτε τις διαφορές στην ταχύτητα εκτέλεσης.