1ο Εργαστήριο Αρχιτεκτονικής Η/Υ: MIPS assembly, γνωριμία με τον MARS

Α. Ευθυμίου

Παραδοτέο: Τρίτη 23 Φλεβάρη, 23:00

Μή ξεχάσετε να επιστρέψετε, μέσω GitHub, το παραδοτέο που αναφέρεται στο τέλος του κειμένου για να πάρετε βαθμό γι'αυτή την εργαστηριακή άσκηση!

Με αυτή την εργαστηριακή άσκηση θα εξοικειωθείτε με τον MARS, έναν προσομοιωτή γλώσσας assembly του MIPS, που θα χρησιμοποιήσετε σε πολλές επόμενες εργαστηριακές ασκήσεις. Θα πρέπει να έχετε μελετήσει το πρώτο μάθημα για τη γλώσσα assembly του MIPS (3η διάλεξη του μαθήματος) που αντιστοιχεί στις ενότητες 2.1-2.3 του βιβλίου.

Δευτερεύων, αλλά πολύ σημαντικός, στόχος της άσκησης είναι η εξοικίωσή σας με τον τρόπο παράδοσης των ασκήσεων του μαθήματος μέσω του GitHub. Το git απαιτεί κάποιο χρόνο εκμάθησης γιατί είναι αρκετά διαφορετικό από ότι, οι περισσότεροι, έχετε συνηθίσει μέχρι τώρα. **Είναι σημαντικό να μην περιμένετε μέχρι την τελευταία στιγμή για να παραδόσετε την άσκηση.** Κάντε μερικές δοκιμές νωρίς ώστε να μπορείτε να αντιμετωπίσετε τυχόν προβλήματα. Δεν χρειάζεται να έχετε ολοκληρώσει την άσκηση για να κάνετε μια δοκιμαστική παράδοση. Μια μικρή αλλαγή σε ένα σχόλιο του προγράμματος είναι αρκετή για να ελέγξετε ότι μπορείτε να παραδόσετε την άσκηση σωστά. Μόνο η τελευταία παράδοση βαθμολογείται.

Χρησιμοποιείστε το Piazza για ανταλλαγή συμβουλών, πληροφοριών, καλών πρακτικών χρήσης κτλ, αλλά μή δίνετε πληροφορίες που φανερώνουν τη λύση των ασκήσεων. Οι βοηθοί και ο διδάσκοντας θα συμμετέχουν στις συζητήσεις αυτές. Στο Piazza υπάρχει μια εκτενής ανάρτηση για τη χρήση του git,GitHub στις εργαστηριακές ασκήσεις του μαθήματος και τα πιο συνηθισμένα προβλήματα που μπορεί να παρουσιαστούν.

Ο MARS, που αναπτύχθηκε από τους Pete Sanderson και Ken Vollmar, εκτός απο προσομοιωτής είναι ένα πλήρες γραφικό περιβάλον ανάπτυξης και εξσφαλμάτωσης προγραμμάτων (IDE). Στο εργαστήριο 0 δόθηκαν οδηγίες για την εγκατάστασή του. Επειδή είναι γραμμένος σε Java τρέχει σε υπολογιστές με οποιοδήποτε λειτουργικό σύστημα αρκεί να υπάρχει εγκατεστημένη η κατάλληλη έκδοση του Java J2SE. Στους υπολογιστές των εργαστηρίων του Τμήματος θα τον βρείτε στο `~myy402/bin/MarsMYY402_4_5.jar`.

1 Βασικές πληροφορίες για MIPS assembly

- Τα προγράμματα assembly αποθηκεύονται σε απλά αρχεία κειμένου και οι καταλήξεις τους είναι συνήθως `.asm` ή `.s`.
- Τα προγράμματα καλό είναι να έχουν μία ετικέτα (**label**) με το όνομα `main:` που δηλώνει από που θα ξεκινήσει η εκτέλεση. Στον MARS χρειάζεται και η δήλωση `.globl main` στην αρχή του προγράμματος. Στις εργαστηριακές ασκήσεις πρέπει απαραίτητα να υπάρχει η ετικέτα main. Δείτε παρακάτω μία ρύθμιση του MARS που πρέπει να κάνετε ώστε τα προγράμματά σας να ξεκινούν από το main.
- Τα προγράμματα πρέπει να τελειώνουν με τις εξής δύο εντολές: `addi $v0,$0,10` και `syscall`. Με αυτές, όταν ολοκληρωθεί η εκτέλεση του προγράμματος, ο έλεγχος περνάει ξανά στο λειτουργικό σύστημα.
- Τα σχόλια ξεκινούν με το σύμβολο # και τελειώνουν στο τέλος της γραμμής.
- Δεν επιτρέπεται να γράψετε πάνω από μία εντολή ανά γραμμή.
- Οι ετικέτες (labels) πρέπει να τελειώνουν με το σύμβολο :
- Ο assembler, το πρόγραμμα που διαβάζει κώδικα assembly, είναι εξαιρετικά απλός. Μπορεί να μην επιτρέπει για παράδειγμα οι εντολές να «σπάνε» σε ξεχωριστές γραμμές και τα μηνύματα

λάθους μπορεί να μην είναι πάντα αρκετά κατατοπιστικά. Ακολουθήστε το πρότυπο των προγραμμάτων που σας δίνονται για να γράφετε σωστό κώδικα που ακολουθεί το στυλ με το οποίο είναι εξοικειωμένοι οι περισσότεροι προγραμματιστές στον κόσμο. Οι βασικές οδηγίες είναι:

  - οι ετικέτες γράφονται τέρμα αριστερά και είναι σχετικά μικρές.
  - οι εντολές ξεκινούν δεξιότερα για να ξεχωρίζουν από τις ετικέτες και είναι στοιχισμένες.
  - οι τελεσταίοι των εντολών επίσης απέχουν μερικά κενά από τα ονόματα των εντολών.

- Ο assembler, εκτός από εντολές assembly, ετικέτες και σχόλια, δέχεται και κάποιες ειδικές λέξεις ως οδηγίες που λέγονται directives. Χρησιμοποιούνται για να ξεχωρίσουν το πρόγραμμα από τα δεδομένα και για να δεσμεύσουν χώρο στη μνήμη για δεδομένα. Τα directives ξεκινούν πάντα με μία τελεία. Θα μάθετε μερικές απαραίτητες directives από το πρώτο πρόγραμμα assembly. Τα σχόλια του προγράμματος εξηγούν τι σημαίνει η κάθε μία.

2 Εξοικείωση με τον MARS

Κάντε τα παρακάτω βήματα:

1. Το repository σας στο GitHub, έχει τη μορφή: <όνομα χρήστη GitHub>-labs. Για την ώρα είναι άδειο. Παρόλα αυτά, κλωνοποιείστε το, στον υπολογιστή σας (εργαστήριο ή σπίτι ή και τα δύο), αγνοώντας το σχετικό μήνυμα του git. Δείτε τις σχετικές οδηγίες για το git από το 2ο μάθημα και το φυλλάδιο του εργαστηρίου 0.

   Για να πάρετε τα αρχεία της εργαστηριακής άσκησης, μεταβείτε στον κατάλογο που θα δημιουργηθεί από το παραπάνω βήμα και θα έχει το ίδιο όνομα με το αποθετήριο (αλλάζετε το <GitHub-userName> με το όνομα χρήστη):

   ```
   cd <GitHub-userName>-labs
   ```

   Μετά, δώστε τις παρακάτω εντολές (προσοχή, η αντιγραφή-επικόλληση από κείμενα pdf συχνά δημιουργεί προβλήματα):

   ```
   git remote add lab01_starter https://github.com/UoI-CSE-MYY402/lab01_starter.git
   git fetch lab01_starter
   git merge lab01_starter/master -m "Fetched lab01 starter files"
   ```

   Θα δείτε ότι θα εμφανιστεί ένας κατάλογος lab01. Μεταβείτε σε αυτόν: cd lab01. Εκεί θα βρείτε το αρχείο `lab01.asm`, το πρώτο σας πρόγραμμα σε MIPS assembly.

2. Ξεκινήστε τον MARS, π.χ. σε τερματικό με την εντολή `java -jar <path_to_MarsMYY402_4_5.jar>`.

3. Στο μενού Settings ενεργοποιήστε το "Initialize Program Counter to global main", ώστε η εκτέλεση να ξεκινά από την ετικέτα main. Επίσης, επειδή δε θα χρειαστείτε τους co-processors, μπορείτε να σείρετε το δεξί υπό-παράθυρο προς τα δεξιά ώστε να φαίνεται μόνο το tab "Registers". Έτσι θα υπάρχει περισσότερος χώρος για να βλέπετε τον κώδικα, κυρίως κατά την εκτέλεσή του (στο tab "execute"). Αν τα γράμματα σας φαίνονται πολύ μικρά, μπορείτε να αλλάξετε το μέγεθός τους (και ό,τι άλλο θέλετε) από τις ρυθμίσεις στο Settings→Editor. Προτείνω τη ρύθμιση "Font Family: Monospaced" και tab size 4.

4. Φορτώστε το αρχείο `lab01.asm` χρησιμοποιώντας είτε το εικονίδιο (2ο από αριστερά) ή από το μενού: File→Open, ή, ακόμα καλύτερα, με τη συντόμευση από το πληκτρολόγιο (Ctrl-O).

   Διαβάστε προσεκτικά τον κώδικα (και τα σχόλια) στο "Edit" tab. Παρατηρήστε ότι οι εντολές, ετικέτες και σχόλια χρωματίζονται αυτόματα (code highlighting) και ότι υπάρχει η δυνατότητα αυτόματης συμπλήρωσης εντολών (completion suggestion).

5. Όταν πλέον καταλαβαίνετε καλά τον κώδικα, ετοιμάστε τον για εκτέλεση (αγγλικός όρος assemble) χρησιμοποιώντας το μενού (Run→Assemble), ή το εικονίδιο με το κατσαβίδι και το κλειδί ή με το πλήκτρο F3.

   Αυτόματα θα αλλάξει ο MARS στο tab "Execute", όπου φαίνονται 3 παράθυρα στη θέση του κώδικα: η μνήμη εντολών (text segment), η μνήμη δεδομένων (data segment) και η αντιστοίχιση ετικετών σε διευθύνσεις (labels). Στα δεξιά θα παραμείνει το παράθυρο με τους καταχωρητές.

   Παρατηρήστε τις πληροφορίες στα παράθυρα. Ο αρχικός κώδικας (και τα σχόλια) φαίνονται ακόμη μαζί με άλλες πληροφορίες όπως η διεύθυνση κάθε εντολής και η κωδικοποίησή της σε γλώσσα μηχανής (που θα είναι το αντικείμενο επόμενου μαθήματος). Θα δείτε ότι η πρώτη γραμμή του κώδικα έχει χρωματιστεί με κίτρινο χρώμα: αυτό υποδεικνύει την εντολή που πρόκειται να εκτελεστεί.

   Στο παράθυρο δεδομένων μπορεί κανείς να δεί τα δεδομένα ως δεκαεξαδικούς αριθμούς, ως δεκαδικούς αριθμούς, ή ως συμβολοσειρα χαρακτήρων με κωδικοποίηση ASCII (1 byte ανά χαρακτήρα). Η μορφή παρουσίασης αλλάζει χρησιμοποιώντας τα tick boxes στο κάτω μέρος του παραθύρου.

   Παρατηρήστε ότι εμφανίζονται 8 λέξεις των 32 bit ανά γραμμή. Η διεύθυνση είναι ο αριθμός στην αρχή της κάθε γραμμής και αντιστοιχεί στην πρώτη λέξη. Η διεύθυνση της πρώτης γραμμής είναι 0x10010000 με τις αρχικές ρυθμίσεις του MARS. Η επόμενη λέξη θα είναι στη διεύθυνση που φαίνεται στην αρχή της γραμμής + 4, δηλαδή 0x10010004. Θυμηθείτε ότι κάθε λέξη είναι 32 bit, δηλαδή 4 byte και η μνήμη δίνει διευθύνσεις σε κάθε byte. Έτσι οι διευθύνσεις διαδοχικών λέξεων διαφέρουν κατά 4. Η διεύθυνση της τελευταίας λέξης της πρώτης γραμμής είναι 0x1001001c, ίση με τη διεύθυνση της πρώτης λέξης $+28_{ten}$ (0x1c). Η απόσταση της διεύθυνσης κάθε λέξης της γραμμής από την αρχική φαίνεται στην επικεφαλίδα της αντίστοιχης στήλης του παραθύρου: είναι ο αριθμός μέσα στην παρένθεση και είναι σε δεκαεξαδική μορφή.

6. Εκτελέστε το πρόγραμμα εντολή προς εντολή χρησιμοποιώντας το Run→Step ή το αντίστοιχο εικονίδιο (πράσινο τρίγωνο με τον αριθμό 1) ή πλήκτρο σύντμευσης. Παρατηρήστε τις αλλαγές στους καταχωρητές και στη μνήμη δεδομένων. Ο MARS δείχνει με χρώμα τις τελευταίες αλλαγές. Σε αντίθεση με το παράθυρο μνήμης εντολών όπου το χρώμα δείχνει την επόμενη εντολή που *θα* εκτελεστεί, στα παράθυρα καταχωρητών και μνήμης το χρώμα δείχνει την αλλαγή που *μόλις έγινε*.

   Σε ένα σημείο της εκτέλεσης σας ζητείται να δώσετε τον αριθμό μητρώου σας (matriculation number). Αυτό γίνεται στο ύπο-παράθυρο (Run I/O) στο κάτω μέρος. Πρέπει να πατήσετε Return / Enter για να δεχθεί την είσοδο ο MARS.

7. Εξερευνήστε όλα τα μενού εκτός του Tools και προσπαθείστε να μάθετε τις συντμεύσεις πλήκτρων. Θα χρησιμοποιείσετε το MARS σε πολλές εργαστηριακές ασκήσεις: όσο περισσότερο γνωρίζετε τις δυνατότητές του τόσο πιο παραγωγικοί θα είστε, συνεπώς θα τελειώνετε τις ασκήσεις γρηγορότερα.

   Ενδεικτικά αναφέρονται οι παρακάτω χρήσιμες λειτουργίες.

   Υπάρχει η δυνατότητα να "πάτε προς τα πίσω" (backstep) μία εντολή κάθε φορά (εικονίδιο: μπλέ τρίγωνο με τον αριθμό 1), να επανεκκινήσετε το πρόγραμμα (reset), καθώς και να καθορίσετε την ταχύτητα εκτέλεσης (slider επάνω δεξιά) ώστε να παρακολουθείτε τις αλλαγές που συμβαίνουν όταν εκτελείτε ένα μεγάλο πρόγραμμα (Run→Go).

   Κατά τη διάρκεια της εκτέλεσης, μπορείτε, εκτός από το να παρατηρείτε τιμές καταχωρητών και μνήμης, να τις αλλάξετε με διπλό κλικ στο πεδίο τιμής (value).

   Κάνοντας κλικ στο όνομα μιας ετικέτας (label) στο παράθυρο ετικετών, εμφανίζεται ένα λεπτό μπλε πλαίσιο στο παράθυρο δεδομένων στη λέξη που αντιστοιχεί στην ετικέτα αυτή.

3 Ερωτήσεις αυτοεξέτασης

Για να βεβαιωθείτε ότι εκτελέσατε σωστά την άσκηση, θα πρέπει να μπορείτε να απαντήσετε τις παρακάτω ερωτήσεις. Προσπαθήστε μόνοι σας. Μη ξεχνάτε ότι ο MARS έχει ένα ενσωματομένο εγχειρίδιο

(μενού Help ή πλήκτρο F1). Αν κολλήσετε, ρωτήστε τους συμφοιτητές σας αυτοπροσώπως ή στο Piazza, αλλά όχι για οτιδήποτε σχετίζεται με το παραδοτέο παρακάτω.

1. Ποιά είναι η διεύθυνση του var3, και ποιά του array+0x10;
2. Πώς μπορείτε να διαβάσετε το string που είναι αποθηκευμένο στο mesg1 στο παράθυρο δεδομένων;
3. Γιατί είναι γραμμένο κάπως ανάποδα;
4. Πώς μπορείτε να αλλάξετε την τιμή της var1 χωρίς να αλλάξετε τον κώδικα από το tab "Edit";
5. Ποιές είναι οι διαφορές των αποτελεσμάτων εκτέλεσης των lb και lbu;

4 Αυτόματος έλεγχος ορθότητας

Στο repository θα βρείτε έναν κατάλογο που λέγεται test. Εκεί μέσα υπάρχει ένας μηχανισμός αυτόματου ελέγχου των αποτελεσμάτων του προγράμματός σας. Ο έλεγχος γίνεται με σύγκριση τιμών καταχωρητών και διευθύνσεων μνήμης με προϋπολογισμένες τιμές που θα πρέπει να δίνει ένα σωστό πρόγραμμα. Αυτό θα χρησιμοποιηθεί και από το διδακτικό προσωπικό ώς ένας από τους τρόπους ελέγχου και βαθμολόγησης.

Για την ώρα το πρόγραμμα που σας δόθηκε δεν περνάει τον έλεγχο. Αν το τρέξετε με το υπάρχον lab01.asm αρχείο, θα πάρετε μηνύματα λάθους γιατί ο ελεγκτής δεν μπορεί να χειριστεί είσοδο από το πληκτρολόγιο. Πρέπει το πρόγραμμα assembly να μπορεί να εκτελεστεί χωρίς διακοπές για είσοδο από το πληκτρολόγιο.

Για τον έλεγχο χρησιμοποιείται η γλώσσα Python 3.4 και, επομένως, αν χρησιμοποιήσετε δικό σας υπολογιστή για την άσκηση, θα πρέπει να την εγκαταστήσετε. Θα χρειαστεί επίσης να αλλάξετε το αρχείο `lab01_tester.py`, στη γραμμή 14, ώστε να μπορεί να βρεί το αρχείο MarsMYY402_4_5.jar. Αντίστοιχες αλλαγές θα πρέπει να κάνετε και για τις επόμενες εργαστηριακές ασκήσεις.

Το αρχείο `mipsTester.py` είναι αυτό που περιέχει τις συναρτήσεις που τρέχουν τον MARS και καταγράφουν την έξοδο της προσομοίωσης και τις τιμές καταχωρητών και μνήμης που μας ενδιαφέρουν. Δε χρειάζεται να κάνετε καμιά αλλαγή σε αυτό.

Το αρχείο `lab01_tester.py` περιέχει τους ελέγχους που θα πραγματοποιηθούν στο πρόγραμμα assembly lab01.asm. Η μεταβλητή tests στη γραμμή 17 του παραπάνω αρχείου είναι μια λίστα από "δοκιμασίες". Κάθε δοκιμασία είναι ένα tuple που αποτελείται από:

- Το όνομα της δοκιμασίας, ένα string.
- Μια λίστα από εκφράσεις, strings, του βοηθητικού προγράμματος sed. Χρησιμοποιείται για να αλλάξει τις τιμές εισόδου ενός προγράμματος. Η συνήθης χρήση είναι να εντοπίσει ένα label και να αλλάξει την τιμή που αποθηκεύεται στη μνήμη στη θέση του label με μία άλλη. Για παράδειγμα, στο `lab01_tester.py`, η έκφραση `"s/^matric:.*$/matric: .word %d/" %(matric)` σημαίνει: βρές τη γραμμή που ξεκινά με τη λέξη matric: και άλλαξε την ώστε να γίνει matric: .word ακολουθουμενη από την τιμή της μεταβλητής matric του `lab01_tester.py`. Η μεταβλητή αυτή ορίζεται στη γραμμή 16 και **θα πρέπει να αλλάξετε την τιμή της με τον δικό σας αριθμό μητρώου**. Θα μπορούσατε να βάλετε τον αριθμό μητρώου απευθείας μέσα στο string, αλλά η παραπάνω έκφραση σας δείχνει τις δυνατότητες που παρέχονται από τον μηχανισμό ελέγχου.

  Το βοηθητικό προγραμμα sed βρίσκεται σε κάθε εγκατάσταση Unix, αλλά και σε υλοποιήσεις του bash σε Windows, π.χ. gitBash, cygwin και τερματικό του Mac. Όπως χρησιμοποιείται στο `mipsTester.py`, αλλάζει επιτόπου το αρχείο assembly, κρατώντας το προηγούμενο περιεχόμενό του ως διαφορετικό αρχείο με την προσθήκη στο τέλος του ονόματος του `_b` ακολουθούμενου από έναν αριθμό. Για παράδειγμα, αν το αρχείο που περιέχει το πρόγραμμα είναι το `lab01.asm`, θα

εμφανιστούν αρχεία της μορφής `lab01.asm_b0`, `lab01.asm_b1`. Τα αρχεία αυτά μπορείτε να τα διαγράψετε. Σε καμία περίπτωση μην τα στείλετε στο GitHub. Μόνο το `lab01.asm` χρειάζεται.

Τέλος, δεν χρειάζεται να ανησυχείτε γιατί τρέχοντας τον αυτόματο έλεγχο το κύριο παραδοτέο σας (σε αυτή την άσκηση το `lab01.asm`) αλλάζει. Μόνο τιμές δεδομένων θα αλλάζουν και, κατά την διόρθωση, το ίδιο ακριβώς θα γίνεται: θα τρέχει ένα αντίστοιχο Python script που θα δίνει τιμές και θα περιμένει να δει τις σωστές απαντησεις.

- Ένα dictionary για τις αναμενόμενες τιμές καταχωρητών. Τα κλειδιά του dictionary θα πρέπει να είναι ονόματα καταχωρητών (τύπου string) και οι τιμές του dictionary είναι οι αναμενόμενες τιμές των καταχωρητών μετά την εκτέλεση του προγράμματος. Για παράδειγμα, στο `lab01_tester.py`, το dictionary για τους καταχωρητές είναι το

  ```
  {'s0' : -1, 's1' : 0x00ff, 's2' : 0x1001002c, 't0' : 0x10010050}
  ```

  που σημαίνει ότι στο τέλος της εκτέλεσης ο καταχωρητής `s0` θα πρέπει να έχει την τιμή -1 (0xffffffff σε δεκαεξαδικό), κ.ο.κ.

- Ένα dictionary για τις αναμενόμενες τιμές θέσεων μνήμης. Είναι παρόμοιο με το dictionary των καταχωρητών μόνο που αντί για ονόματα καταχωρητών μπορούν να χρησιμοποιηθουν διευθύνσεις μνήμης ή και ετικέτες μνήμης.

Η τελευταία γραμμή του `lab01_tester.py` τρέχει όλες τις δοκιμασίες και εμφανίζει τα αποτελέσματα. Αν υπάρχουν λάθη, μπορεί κανείς να δεί περισσότερες πληροφορίες αλλάζοντας το verbose σε True.

Ως μέρος του παραδοτέου αυτής της άσκησης θα αλλάξετε την τιμή της μεταβλητής matric ώστε να είναι ο δικό σας αριθμός μητρώου.

5 Παραδοτέο

Αλλάξτε τις γραμμές 23-34 του `lab01.asm` ώστε, αντί να ζητάει να διαβάσει από το πληκτρολόγιο τον αριθμό μητρώου, να τον φορτώνει από τη μνήμη στον καταχωρητή `v0` με μία ή δύο κατάλληλες εντολές. Αποθηκεύστε τον αριθμό μητρώου σας στη μνήμη γράφοντάς τον στη θέση της ετικέτας matric: χρησιμοποιώντας τον editor του MARS, αντικαταστήστε το 0 με τον αριθμό μητρώου σας στην παραπάνω ετικέτα. Μη ξεχάσετε να ξανακάνετε assemble το πρόγραμμά σας αφού τώρα πια έχει αλλάξει.

Το αλλαγμένο πρόγραμμα δεν θα εμφανίζει κάτι στην οθόνη, ούτε θα ζητάει είσοδο από τον χρήστη. **Γενικά στην τελική έκδοση των προγραμμάτων που θα παραδίδετε μην αφήνετε κώδικα που εμφανίζει πληροφορίες στην οθόνη γιατί υπάρχει κίνδυνος να μπερδευτεί το Python script που κάνει τον αυτόματο έλεγχο.**

Αφού δοκιμάσετε ότι τρέχει σωστά, αλλάξτε τις αναμενόμενες τιμές στο αρχείο `test/lab01_tester.py` (την τιμή της μεταβλητής matric) και τρέξτε το από τον κατάλογο test, χρησιμοποιώντας ένα τερματικό.

Μη ξεχάσετε να κάνετε commit τις αλλαγές σας και να τις στείλετε (push) στο GitHub repository για να βαθμολογηθούν πριν από την καταληκτική ημερομηνία!