3ο Εργαστήριο Αρχιτεκτονικής Η/Υ: MIPS assembly: έλεγχος αν ένα string είναι παλίνδρομο με αναδρομική υπορουτίνα

Α. Ευθυμίου

Παραδοτέο: Παρασκευή 12 Νοέμβρη, 23:59

Το αντικείμενο αυτής της άσκησης είναι μια αναδρομική υπορουτίνα που εξετάζει αν ένα string είναι παλίνδρομο. Θα γράψετε ένα προγράμμα assembly που χρησιμοποιεί υπορουτίνες, μία από τις οποίες θα είναι αναδρομική. Θα πρέπει να έχετε μελετήσει τα μαθήματα για τη γλώσσα assembly του MIPS (μέχρι και της Τετάρτης 27/10) που αντιστοιχούν μέχρι την ενότητα 2.9 του βιβλίου (διάλεξη 7).

Με την άσκηση αυτή θα μάθετε να γράφετε υπορουτίνες και να χρησιμοποιείτε τη στοίβα του συστήματος.

1 Παλίνδρομα

Μια λέξη είναι παλίνδρομη όταν διαβάζεται ίδια και κανονικά και ανάποδα. Για παράδειγμα τα παρακάτω string είναι παλίνδρομα: bob, anna.

Γενικά αν ένα string S παριστάνεται ως S = $S_1 S_2 \dots S_N$, όπου N το μήκος του string, το S είναι παλίνδρομο, αν $N = 0$ ή $N = 1$ ή $S_1 = S_N$ και το υπο-string $(S_2 \dots S_{N-1})$ είναι παλίνδρομο.

Ο σκοπός της άσκησης είναι η συγγραφή μια υπορουτίνας σε MIPS assembly που θα ελέγχει αν ένα string είναι παλίνδρομο. Ο παραπάνω ορισμός είναι αναδρομικός και ο αλγόριθμος που θα εφαρμόσετε θα πρέπει επίσης να είναι αναδρομικός. Η υπορουτίνα θα επιστρέφει την τιμή 1 αν το string είναι παλίνδρομο, αλλιώς θα επιστρέφει την τιμή 0.

Η υπορουτίνα θα έχει ως εισόδους τη διεύθυνση του πρώτου χαρακτήρα του string, `$a0` και το μήκος του - πλήθος των χαρακτήρων, `$a1`. Επειδή η σύμβαση είναι τα string να τελειώνουν με τον αριθμό 0 (`'\0'`), αυτός ο αριθμός **δεν** προσμετράται στο μήκος του string. Στις αναδρομικές κλήσεις, όπου εξετάζεται ένα υπο-string, προφανώς τα `$a0`, `$a1` προσαρμόζονται όπως περιγράφει ο παραπάνω ορισμός. Συμφωνα με τον παραπάνω ορισμό το κενό string είναι παλίνδρομο και κατ'επέκταση το ίδιο θα ισχύει αν ως διεύθυνση του string δοθεί το 0 - null.

Προσοχή η άσκηση απαιτεί αναδρομή και όχι επανάληψη στην υλοποίηση της υπορουτίνας. Υπορουτίνες που δεν είναι αναδρομικές ή χρησιμοποιούν άλλο αλγόριθμο θα βαθμολογηθούν με χαμηλό βαθμό ακόμα και αν ειναι σωστές.

2 Η άσκηση

Για να ξεκινήσετε, ακολουθήστε τον σύνδεσμο https://classroom.github.com/a/TBC. Κάνοντας κλικ στον σύνδεσμο, δημιουργείται ένα νέο αποθετήριο στον οργανισμό του μαθήματος. Μπορείτε να δείτε το νέο αποθετήριο αμέσως μετά το κλίκ στον σύνδεσμο.Το URL του αποθετηρίου θα έχει τη μορφή https://github.com/UoI-CSE-MYY505/lab03-ghUsername, όπου ghUsername το όνομα χρήστη που έχετε στο GitHub. Κλωνοποιήστε το για να πάρετε τα αρχεία της εργαστηριακής άσκησης και μεταβείτε στον κατάλογο που θα δημιουργηθεί από το παραπάνω βήμα και θα έχει το ίδιο όνομα με το αποθετήριο.

Στο αρχείο `lab03.asm` θα βρείτε έναν μικρό σκελετό του κώδικα με ένα παράδειγμα εισόδου και μια κλήση στην υπορουτίνα και σε σχόλια θα δείτε που θα πρέπει να γράψετε την υπορουτίνα σας.

Στην άσκηση αυτή είναι πολύ σημαντικό να ακολουθηθεί η σύμβαση για τη χρήση των καταχωρητών του MIPS που εξηγήθηκε στο μάθημα. Το πιο συνηθισμένο λάθος είναι η θεώρηση ότι κάποιοι καταχωρητές (t, a, v, ra) διατηρούνται μετά από μια κλήση υπορουτίνας. Θα πρέπει να θεωρείτε ότι κάθε φορά που εκτελείται μία jal, οι τιμές όλων των καταχωρητών με όνομα που ξεκινάει από t, a, v και ο ra έχουν αλλοιωθεί. Συνεπώς θα πρέπει να αποθηκεύετε, όποιους από αυτούς τους καταχωρητές χρειάζεστε, στην στοίβα πριν την εντολή jal. Ένα άλλο συνηθισμένο λάθος είναι το πέρασμα τιμών σε υπορουτίνες

μέσω των καταχωρητών s σαν να ήταν καθολικές (global) μεταβλητές. Τέλος, μην αλλάξετε την σειρά των καταχωρητών που περνούν παραμέτρους εισόδου ή εξόδου. Αν μια υπορουτίνα έχει μία είσοδο, τότε υποχρεωτικά χρησιμοποιείται ο a0. Παρόμοια για τις εξόδους - τιμές επιστροφής: αν υπάρχει μία έξοδος χρησιμοποιείται υποχρεωτικά ο v0. Μπορεί κανείς πράγματι να γλιτώσει αρκετές εντολές παρακάμπτοντας αυτούς τους περιορισμούς, αλλά δεν πρέπει να το κάνετε!

3 Stack visualizer - εργαλείο Mars

Όταν χρησιμοποιούνται υπορουτίνες, ειδικά αναδρομικές, είναι χρήσιμο να μπορεί κανείς να δει τί γίνεται στη στοίβα. Μπορείτε να δείτε τα περιεχόμενα της μνήμης που αντιστοιχούν στη στοίβα, επιλέγοντας το current sp στο (υπο)παράθυρο data segment στο execute tab του MARS.

Αλλά υπάρχει καλύτερος τρόπος! Δύο απόφοιτοι, πλέον, του Τμήματος πρόσθεσαν το εργαλείο Stack Visualizer στον Mars. Θα το βρείτε στο μενού Tools. Αφού το ανοίξετε, πατήστε connect to MIPS. Μπορεί να ζητηθεί να ξαναγίνει assemble το πρόγραμμα. Μετά σας δείχνει με παραστατικό τρόπο την εικόνα της στοίβας, την τρέχουσα θέση του $sp, καθώς και τον καταχωρητή που αποθήκευσε μια υπορουτίνα στη στοίβα, μαζί με το όνομα της υπορουτίνας.

4 Παραδοτέο και κριτήρια αξιολόγησης

Το παραδοτέο της άσκησης είναι το αρχείο lab03.asm που περιέχει το πρόγραμμά σας. Μην αλλάξετε το όνομα του αρχείου, γιατί δεν θα το βρίσκει ο αυτόματος έλεγχος! Προεραιτικά αλλάξτε και το README.md.

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

Τα προγράμματά σας θα βαθμολογηθούν για την ορθότητά τους, τη χρήση αναδρομικού αλγόριθμου, την συμμόρφωση με το πρότυπο χρήσης καταχωρητών, την ποιότητα σχολίων και τη ταχύτητα εκτέλεσής τους.