3ο Εργαστήριο Αρχιτεκτονικής Η/Υ: MIPS assembly: Αναδρομική δυαδική αναζήτηση σε πίνακα δεικτών σε strings

Α. Ευθυμίου

Παραδοτέο: Παρασκευή 6 Νοέμβρη, 23:59

Το αντικείμενο αυτής της άσκησης είναι ένα πρόγραμμα που αναζητά ένα string σε έναν ταξινομημένο πίνακα δεικτών strings με αναδρομική δυαδική αναζήτηση. 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 Πέμπτης 22/10) που αντιστοιχούν μέχρι την ενότητα 2.8 του βιβλίου (διάλεξη 7).

Δυαδική αναζήτηση υλοποιήθηκε στην προηγούμενη άσκηση σε επαναληπτική μορφή και με ακεραίους ως στοιχεία του πίνακα. Αυτή την φορά ο πίνακας θα περιέχει δείκτες σε strings επομένως η σύγκριση θα είναι πιο σύνθετη και θα υλοποιείται με μια υπορουτίνα (strcmp). Επιπλέον η δυαδική αναζήτηση θα γίνεται με αναδρομικό τρόπο αυτή τη φορά.

1 Η άσκηση

Για να ξεκινήσετε, ακολουθήστε τον σύνδεσμο https://classroom.github.com/a/. 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. Υλοποιείστε και ελέγξτε πρώτα την υπορουτίνα σύγκρισης string (strcmp) και μετά συνεχίστε στην υλοποίηση της αναδρομικής δυαδικής αναζήτησης.

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) 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 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!

1.1 Σύγκριση string

Η υπορουτίνα σύγκρισης string δέχεται στους καταχωρητές a0, a1 τις διευθύνσεις των strings (διευθύνσεις του πρώτου χαρακτήρα κάθε string). Το αποτέλεσμα, στον v0 θα είναι 0 αν είναι ίσα, ένας αρνητικός αριθμός, αν το string με διεύθυνση a0 είναι "μικρότερο", και θετικός αριθμός στην αντίθετη περίπτωση. Οι τιμές των αρνητικών ή θετικών αριθμών δεν είναι προκαθορισμένες. Μπορείτε να χρησιμοποιήσετε ό,τι βολεύει την υλοποίηση της υπορουτίνας.

Η σύγκριση δύο strings γίνεται ανά χαρακτήρα συγκρίνοντας τις τιμές ASCII. Δεν γίνεται κάποια διάκριση για κεφαλαία ή μικρά. Ό,τι δίνει η σύγκριση των κωδικών ASCII, αυτό χρησιμοποιείται ως το αποτέλεσμα σύγκρισης των χαρακτήρων. Έτσι το "aris" θα είναι μετά το "Zachary", γιατί το μικρό a είναι μετά το κεφαλαίο Z και η υπορουτίνα θα εστρέφει θετικό αριθμό (με το "aris" στον a0). Αν οι

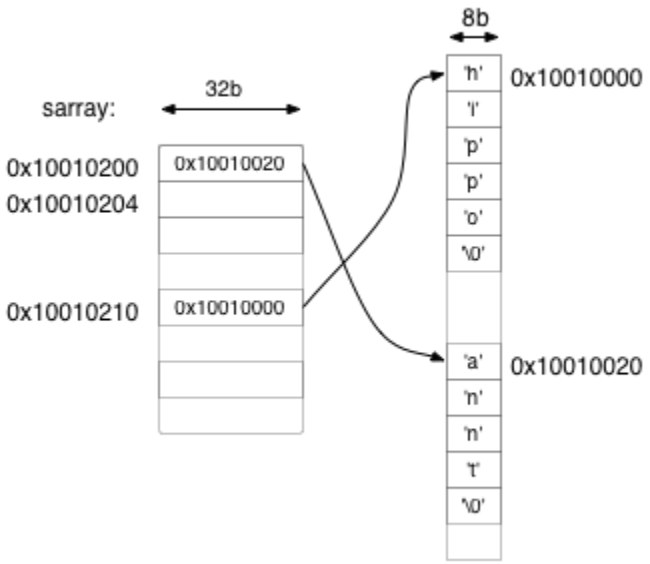
χαρακτήρες είναι ίδιοι, η σύγκριση προχωράει στον επόμενο κ.ο.κ. μέχρι το τέλος του μικρότερου από τα δύο. Για να είναι ακριβώς ίδια δύο string, πρέπει να έχουν όλους τους χαρακτήρες τους ίδιους, και, φυσικά, να έχουν το ίδιο μήκος - πλήθος χαρακτήρων.

1.2 Αναδρομική δυαδική αναζήτηση

Η υπορουτίνα αυτή (`rec_b_search`) θα δέχεται ως εισόδους την αρχική διεύθυνση του πίνακα (`a0`), την τελική διεύθυνση του πίνακα (`a1`), όπου θα είναι αποθηκευμένο το τελευταίο στοιχείο του πίνακα, και έναν δείκτη-διεύθυνση του string που αναζητείται (`a2`). Έτσι η αναδρομική κλήση θα περνάει έναν δείκτη στο πρώτο στοιχείο, έναν δείκτη στο τελευταίο και τον `a2` που είναι η διεύθυνση του ζητούμενου string. Κάθε φορά είτε ο αρχικός δείκτης είτε ο τελκός θα αλλάζουν ανάλογα με το αποτέλεσμα της σύγκρισης του "μεσαίου" στοιχείου.

Ο πίνακας θα περιέχει δείκτες σε strings. Στο σχήμα φαίνεται ένα παράδειγμα. Το πρώτο στοιχείο είναι ο δείκτης στο string "ant" που βρίσκεται στη διεύθυνση 0x10010020, ενώ κάπου παρακάτω βρίσκεται ο δείκτης στο string "hippo" που βρίσκεται στη διεύθυνση 0x10010000. Στο σχήμα φαίνονται και οι διευθύνσεις των πρώτων στοιχείων του πίνακα.

Η υπορουτίνα rec_b_search θα επιστρέφει στον `v0` την διεύθυνση του δείκτη string που βρέθηκε ή 0 (null pointer) αν δεν βρέθηκε το string. Στο παράδειγμα του σχήματος, θα επιστρέφεται η τιμή 0x10010210 αν αναζητείται το "hippo", γιατί αυτή είναι η διεύθυνση του στοιχείου του πίνακα που δείχνει στο "hippo".



Σχήμα 1: Παράδειγμα πίνακα δεικτών strings.

Το αρχείο `Lab03TestRecBSearch.java` περιέχει τεστ και για την strcmp και για την rec_b_search. Δεν χρειάζεται να κάνετε αλλαγές σε αυτό. Για να το τρέξετε, δώστε τις εντολές:

```
javac -cp munit.jar Lab03TestRecBSearch.java
java -jar munit.jar lab03.asm Lab03TestRecBSearch.class
```

2 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, την συμμόρφωση με το πρότυπο χρήσης καταχωρητών, την ποιότητα σχολίων και τη ταχύτητα εκτέλεσής τους.