



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΠΕΞΕΡΓΑΣΤΩΝ & ΥΛΙΚΟΥ

ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΤΟ ΜΑΘΗΜΑ:

ΗΡΥ 201 ΨΗΦΙΑΚΟΙ ΥΠΟΛΟΓΙΣΤΕΣ

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2020-2021

Εργαστήριο 3:

Απλές διαδικασίες (υπορουτίνες), Βρόχοι, Πράξεις επί Bytes μιας λέξης, επεξεργασία συμβόλων

ΕΚΠΟΝΗΣΗ: Δρ. Ε. Σωτηριάδης, Δρ. Α. Ιωάννου

Α. Σκοπός του εργαστηρίου

Όταν κάνετε ένα ερώτημα στο Google, έχει νόημα κάποιοι χαρακτήρες να αγνοούνται, και όλοι οι χαρακτήρες να μετατρέπονται σε ένα είδος (π.χ. όλα τα γράμματα κεφαλαία ή όλα μικρά). Σε αυτό το εργαστήριο θα δούμε ψύγμα πραγματικού κώδικα που τρέχει σε όλους τους υπολογιστές, μαθαίνοντας ταυτόχρονα καλύτερα την Assembly του MIPS.

Σκοπός του εργαστηρίου είναι η υλοποίηση βρόχων, προσπέλαση μνήμης (memory addressing), απλή χρήση διαδικασιών (υπορουτινών – procedure calls) αλλά χωρίς χρήση της στοίβας είτε για ορίσματα είτε για διεύθυνση επιστροφής), επεξεργασία σε επίπεδο Byte και χρήση λογικών πράξεων.

Στο εργαστήριο αυτό θα μετατρέψουμε όλα τα γράμματα της συμβολοσειράς εισόδου σε μικρά, θα αφήσουμε τους αριθμούς ως έχουν, θα αφαιρέσουμε άλλους χαρακτήρες, και όσον αφορά τα κενά (spaces) θα αφήνουμε ακριβώς ένα κενό ανάμεσα σε λέξεις (ακόμη και αν ξεχώριζαν με άλλους χαρακτήρες).

Ενδεικτικές εισοδοί/έξοδοι παρατίθενται παρακάτω:

Είσοδος	Έξοδος
input with numbers 123 and 456\n	input with numbers 123 and 456\n
INpUt WiTh numBERS 123 and 456 \n	input with numbers 123 and 456\n
input%with\$numbers %^\$^\$^123()*(**())and456*((&* \n	input with numbers 123 and 456\n
INPUT&WITH*NUMBERS)))123 AND 456 *(* \n	input with numbers 123 and 456\n
inpuT WiTh)(*)(*nUmBeRs (*(123)and(456 \n	input with numbers 123 and 456\n

B. Περιγραφή της λειτουργίας του κώδικα

Η ιδέα είναι απλή, δεσμεύουμε στην μνήμη δύο χώρους μνήμης από 100 Bytes έκαστος με την οδηγία **.space**. Όλοι οι χαρακτήρες είναι από το Αγγλικό πληκτρολόγιο ASCII. Δίνουμε στον χρήστη "Please Enter your String:", και ο χρήστης δίνει μία συμβολοσειρά (string) που πάντα τελειώνει με Return και αποτελείται από οποιοδήποτε συνδυασμό των παρακάτω: (α) κεφαλαίους χαρακτήρες, (β) μικρούς χαρακτήρες, (γ) κενά και άλλα σύμβολα όπως &^%\$, αλλά όχι control χαρακτήρες, και τελειώνει κανονικά με \n, μέχρι 100 χαρακτήρες μαζί με το \n. Το πρόγραμμα δεν πανωγράφει τον αρχικό χώρο που έχει γραφτεί η συμβολοσειρά, αλλά μεταφέρει μετά την επεξεργασία την αρχική συμβολοσειρά στον δεύτερο δεσμευμένο χώρο μνήμης, μετατρέποντας όλα τα κεφαλαία γράμματα σε μικρά, αφήνοντας μικρά γράμματα και αριθμούς ως έχουν, και αφαιρώντας ότι δεν είναι αλφανουμερικοί χαρακτήρες. Θεωρούμε ότι οι λέξεις στην είσοδο ξεχωρίζουν με τον χαρακτήρα space ή οποιοδήποτε μη αλφανουμερικό χαρακτήρα, και μπορούν τέτοιοι χαρακτήρες να είναι πολλοί ανάμεσα σε λέξεις. Για την έξοδο οι λέξεις θα ξεχωρίζουν με ακριβώς ένα space. Το αποτέλεσμα πρέπει να είναι μία συμβολοσειρά στην δεύτερη μνήμη που δεσμεύσαμε, και αφού ολοκληρωθεί η διαδικασία μετατροπής να εκτυπωθεί στην κονσόλα. Εφόσον δεν μεσολαβεί χαρακτήρας που διαχωρίζει λέξεις, μία λέξη μπορεί να περιέχει γράμματα και αριθμούς, π.χ. το "INT32" θα γίνει "int32" αλλά όχι "int 32".

Για το πρόγραμμα αυτό επιτρέπεται να χρησιμοποιήσουμε την εντολή SB παρότι οδηγεί σε ιδιαίτερα αναποτελεσματικό κώδικα (υπερβολικά πολλές προσπελάσεις στην μνήμη), αλλά όχι την εντολή LB – τα Load πρέπει να γίνουν με LW, όπως θα εξηγηθεί στο Φροντιστήριο. Αυτό σημαίνει ότι πρέπει να κάνετε ένα βρόχο while διαβάζοντας λέξη-λέξη αλλά γράφοντας χαρακτήρα-χαρακτήρα, βάζοντας και το \n στο τέλος. Προσοχή στο ότι η συμβολοσειρά εισόδου δεν είναι απαραίτητο να έχει ακέραιο αριθμό λέξεων, δηλαδή να έχει αριθμό χαρακτήρων που να είναι πολλαπλάσιο του 4, μπορεί π.χ. να έχει 12 ή 43 ή 57 χαρακτήρες, δηλαδή ο συνολικός αριθμός χαρακτήρων δεν θα είναι απαραίτητα πολλαπλάσιο του 4.

Γ. Υλοποίηση

Να υλοποιήσετε ένα πρόγραμμα σε MIPS Assembly το οποίο υλοποιεί τις παραπάνω λειτουργίες, και ως προς τον χρήστη φαίνεται στην κονσόλα ως εξής:

Please Enter your String:

(π.χ. **INPut (*& WITH\$numBERS 123 and 456 \n)**

The Processed String is:

(π.χ. **input with numbers 123 and 456\n)**

Κατόπιν το πρόγραμμα μπορεί να τερματίζει ή να επανέρχεται στο “Please Enter your String” σε ατέρμονα βρόχο – δική σας επιλογή. Για το πρόγραμμα αυτό ζητούνται τα εξής:

1. Χρήση διαδικασιών (υπορουτινών – procedure calls). Το κυρίως πρόγραμμα έχει τρεις διαδικασίες, μία `Get_Input` για το prompt και το διάβασμα της εισόδου, μία `Process` για την δημιουργία της συμβολοσειράς εξόδου, και μία `Print_Output` για την εκτύπωση του μηνύματος εξόδου και της ίδιας της εξόδου. Για τώρα είναι OK να μην περνάτε ορίσματα όπως π.χ. η διεύθυνση όπου αποθηκεύονται οι συμβολοσειρές (τις αντιμετωπίζουμε σαν `global variables` που είναι γνωστές τόσο στο πρόγραμμα όσο και στις υπορουτίνες). Επίσης δεν χρειάζεται για το εργαστήριο αυτό και μόνο να χρησιμοποιήσουμε στοίβα για αποθήκευση/αποκατάσταση της διεύθυνσης επιστροφής των διαδικασιών επειδή δεν έχουμε φωλιασμένες υπορουτίνες και επομένως ο καταχωρητής `$ra` δεν πανωγράφεται.

2. Χρήση βρόχων `while`. Θεωρούμε ότι πάντα η συμβολοσειρά θα είναι έως 100 χαρακτήρες μαζί με το `\n`, ότι πάντα θα τελειώνει σε `\n`, και ότι πάντα θα έχει αλφανεμερικούς χαρακτήρες και σύμβολα αλλά όχι `control characters`. Δεν ξέρουμε όμως το μέγεθος κάθε φορά, και επειδή φέρνουμε μία λέξη την φορά από την μνήμη το `\n` μπορεί να είναι στο `byte 0 – byte 3` της λέξης. Με πιο ρεαλιστικές παραδοχές θα χρειαζόταν πιο πολύπλοκος κώδικας για να μην μας ξεφύγει ο `pointer` στη μνήμη, αλλά για τώρα η απλούστευση αρκεί.

3. Χρήση εντολών `LW`, `SB` για να φέρνουμε /αποθηκεύουμε 4 bytes από την/στην μνήμη. Χρήση κατάλληλων εντολών για να κάνουμε πράξεις στα σωστά bytes κάθε φορά (π.χ. εντολές δεξιάς ολίσθησης κατά 8 θέσεις για να έρχεται στο «κάτω» μέρος του καταχωρητή ένα `byte` τη φορά, λογικές εντολές όπως `AND/ANDI` για να μασκάρουμε το υπόλοιπο της λέξης, κλπ. Η επεξεργασμένη συμβολοσειρά εξόδου εκτυπώνεται με ένα `syscall 4`.

Δ. Παραδοτέα – Βαθμολογία

Ένα `.zip` αρχείο που περιλαμβάνει:

- A) ένα `.pdf` αρχείο με screenshots αποτελέσματος των δυνατών περιπτώσεων που δείχνει την καλή λειτουργία του συστήματος
- B) το αρχείο `.asm` με τον πηγαίο κώδικα

Σημειώσεις :

1. Πρέπει να έχετε εξαιρετικά καλά σχόλια στον κώδικα, σε κάθε υπορουτίνα να έχετε σχόλια στην αρχή για το ποιους καταχωρητές χρησιμοποιείτε και πως, να κάνετε σωστή χρήση καταχωρητών, κλπ. (20% του βαθμού του εργαστηρίου).

2. Αν χρησιμοποιήσετε `LB` ο βαθμός του εργαστηρίου αυτόματα θα πολλαπλασιαστεί επί 0,5.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ