

Μάθημα: «Οργάνωση Συστημάτων Υπολογιστών»  
Χειμερινό Εξάμηνο 2021-2022

1η Εργασία

Προθεσμία Παράδοσης: 21/11/2021

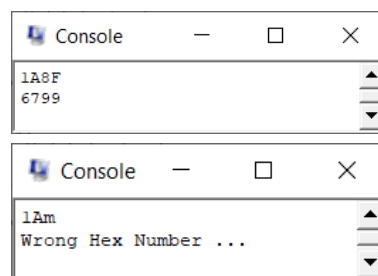
Υπεύθυνη Εργασιών: Μαρία Τογαντζή ([mst@aub.gr](mailto:mst@aub.gr))

Περιγραφή της εργασίας:

Γράψτε ένα πρόγραμμα, στη συμβολική γλώσσα του επεξεργαστή **MIPS32**, που διαβάζει τέσσερις 16-αδικούς χαρακτήρες που παριστάνουν έναν 16-αδικό μη προσημασμένο ακέραιο αριθμό, πυκνώνει τους χαρακτήρες αυτούς στον καταχωρητή **\$t1**, υπολογίζει τον ισοδύναμο δεκαδικό ακέραιο αριθμό και τον εμφανίζει στην οθόνη.

Για παράδειγμα, αν δοθούν από το χρήστη του προγράμματος, ως είσοδος, οι 16-αδικοί χαρακτήρες: **1A8F**

Τότε το πρόγραμμα θα πρέπει να εμφανίσει τον ισοδύναμο δεκαδικό ακέραιο αριθμό: **6799**



Υπόδειξη για τη λύση:

A. Είσοδος και επεξεργασία δεδομένων:

Το πρόγραμμα διαβάζει έναν – έναν τους τέσσερις 16-αδικούς χαρακτήρες (system call 12). Μετά την ανάγνωση κάθε χαρακτήρα, ο τακτικός αριθμός του χαρακτήρα (ascii code) μεταφέρεται στον καταχωρητή \$v0. Έτσι με βάση το παράδειγμα οι διαδοχικές τιμές του καταχωρητή \$v0, θα είναι οι εξής:

Hexadecimal Character	Ascii Code	\$v0
'1'	49	00110001
'A'	65	01000001
'8'	56	00111000
'F'	70	01000110

Ταυτόχρονα, το πρόγραμμα πυκνώνει (**pack**) κάθε χαρακτήρα που διαβάζει (\$v0) στον καταχωρητή **\$t1** (αυτό γίνεται με τις εντολές **or** και **sll** - βλέπε το 2<sup>ο</sup> μέρος του φροντιστηρίου). Έτσι, για το συγκεκριμένο παράδειγμα, το περιεχόμενο του **\$t1**, μετά την ανάγνωση και την πυκνωση των τεσσάρων 16-αδικών θα είναι:

**\$t1:** 00110001 01000001 00111000 01000110 (1A8F<sub>HEX</sub>)

## B. Υπολογισμός και έξοδος αποτελέσματος:

Για κάθε χαρακτήρα (byte) του καταχωρητή \$t1 κάνε (συνολικά 4 φορές):

- Απομόνωσε το **λιγότερο σημαντικό χαρακτήρα (byte)**, δηλαδή από το bit 0 ως το 7 του καταχωρητή \$t1 στον καταχωρητή \$t2 με χρήση της εντολής **and** (βλέπε σημειώσεις φροντιστηρίου).
- Υπολόγισε το δεκαδικό αριθμό (Decimal Number)** που αντιστοιχεί στο 16-αδικό χαρακτήρα (Hexadecimal Character) που βρίσκεται στον καταχωρητή \$t2 ως εξής:

Hexadecimal Character (\$t2)	Ascii Code	Binary	Decimal Number
'1'	49	00110001	49 - 48 = <b>1</b>
'A'	65	01000001	65 - 55 = <b>10</b>
'8'	56	00111000	56 - 48 = <b>8</b>
'F'	70	01000110	70 - 55 = <b>15</b>

Εφόσον ο χαρακτήρας στον \$t2 αντιστοιχεί σε ψηφίο ('0'..'9') να αφαιρέσεις το 48, διαφορετικά εφόσον ο χαρακτήρας είναι γράμμα ('A'..'F') να αφαιρέσεις το 55. Αν ο χαρακτήρας δεν αντιστοιχεί σε 16-αδικό ψηφίο ή γράμμα, τότε το πρόγραμμα θα πρέπει να εμφανίζει ένα κατάλληλο μήνυμα και να τερματίζει τη λειτουργία του. Λάβετε υπόψη ότι οι χαρακτήρες που αντιστοιχούν σε 16-αδικό ψηφίο ('0'..'9') έχουν ascii code 48 ως 57 και οι χαρακτήρες που αντιστοιχούν 16-αδικό γράμμα ('A'..'F') έχουν ascii code 65 ως 70.

- Πολλαπλασίασε** – κάθε φορά – τον αριθμό που προκύπτει από το προηγούμενο βήμα (Decimal Number) **με την κατάλληλη δύναμη του 16** και **άθροισε** το αποτέλεσμα του πολλαπλασιασμού στον αθροιστή που θα περιέχει τον τελικό δεκαδικό ακέραιο:

$$\text{Άθροισμα} = (1 \times 16^3) + (10 * 16^2) + (8 * 16^1) + (15 * 16^0)$$

- Κάνε **λογική μετατόπιση** του \$t1 δεξιά (**srl**) κατά 8 bit (1 byte).

Επανάλαβε για τον επόμενο χαρακτήρα.

Εμφάνισε το αποτέλεσμα (άθροισμα) που για το συγκεκριμένο παράδειγμα είναι ο ακέραιος **6799**.

### **Υποβολή Εργασιών:**

- Η εργασία είναι ατομική.
- Χρησιμοποιείτε το αρχείο ψευδοκώδικα που σας δίνεται και συμπληρώστε τον αντίστοιχο κώδικα στη συμβολική γλώσσα του επεξεργαστή MIPS32.
- Ονομάστε το αρχείο που περιέχει το πρόγραμμα, με τον αριθμό του φοιτητικού σας μητρώου και κατάληξη .s
- Εκτελέστε το πρόγραμμά σας στον προσομοιωτή SPIM και βεβαιωθείτε ότι δεν έχει συντακτικά ή άλλα σφάλματα.
- Τα σχόλια θα πρέπει να είναι γραμμένα με λατινικούς χαρακτήρες (όχι Ελληνικά).
- Συμπίεστε το αρχείο που περιέχει το πρόγραμμα (αριθμός\_μητρώου.s). Υποβάλετε το συμπιεσμένο αρχείο στο eclass το αργότερο μέχρι και **21/11/2021**.
- Τέλος, υπενθυμίζεται, ότι η υλοποίηση της 1ης εργασίας τεκμηριώνει δικαίωμα συμμετοχής στις άλλες δυο (2<sup>η</sup> και 3<sup>η</sup>).

### **Επικοινωνία**

Για τυχόν απορίες σας, μπορείτε να επικοινωνήσετε με την κα Μ. Τογαντζή ([mst@aueb.gr](mailto:mst@aueb.gr)).

```

#      Computer Systems Organization
#      Winter Semester 2021-2022
#      1st Assignment
#
#      Pseudocode by MARIA TOGANTZH (mst@aueb.gr)
#
#      MIPS Code by _____
#      (Please note your name, e-mail and student id number)

        .text
        .globl __start

# ----- Read and Validate Data -----

__start:                                # counter = 4
#
loop:                                   # while counter != 0 do
#
#       Read hex character in $v0
#
#       if $v0 is not ('0'..'9') and is not ('A'..'F') then
#           goto exit_on_error
#       else
#           goto isHex
#
#
isHex:                                  #       shift left $t1
#                                     #       pack $v0 to $t1
#                                     #       counter = counter - 1
#                                     #
#                                     # goto loop

# ----- Calculate Decimal Number -----

exit_loop:                              # print '\n'
#
# result = 0
#
# counter = 4
#
# power = 1
#
# $s1 = 255 (mask = 11111111)
#
loop2:                                  # while counter != 0 do
#
#       $t2 = least significant byte from $t1 - unpack
#       shift right $t1
#
#       if $t2 is letter A..F then
#           $t2 = $t2 - 55
#       else
#           $t2 = $t2 - 48
#
#       $t2 = $t2 * power
#       power = power * 16
#       counter = counter - 1
#       result = result + $t2
#
# goto loop2

```

```

# ----- Print Results -----

exit_loop2:                # print result
                           #
                           # goto exit
                           #
exit_on_error:             #
                           #
                           # print '\n'
                           #
                           # print error message
                           #
                           #
exit:                      #
                           # print '\n'
                           #
                           #
                           # exit
                           #

.data

error: .asciiz "Wrong Hex Number ..."

```