



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
www.cslab.ece.ntua.gr

1η ΑΣΚΗΣΗ ΣΤΗΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ

Ακ. έτος 2018-2019, 5ο Εξάμηνο, Σχολή ΗΜ&ΜΥ

Τελική Ημερομηνία Παράδοσης: **18/11/2018**

ΜΕΡΟΣ Α

Δίνεται το παρακάτω πρόγραμμα σε C καθώς και μια μετάφραση του σε assembly MIPS. Θεωρήστε ότι ο καταχωρητής \$s0 περιέχει τη διεύθυνση του πρώτου στοιχείου του πίνακα array. Συμπληρώστε τα κενά. Σας υπενθυμίζουμε ότι ο καταχωρητής \$0 (ή \$zero) είναι πάντα μηδέν.

```
int array[100];
int *p, N;

p = &array[8];

while (*p != 0) {
    if (*p < 100) *p = *p % N;
    else *p = *p / N;
    p++;
}

                                addi ____, $s0, ____
LOOP:                            lw   ____, 0($s1)
                                ____ $t0, ____, ____
                                div  ____, $s2
                                ____ ____, $t0, 100
                                ____ $t1, ____, ____
                                ____ $t0
                                jmp  ____
ELSE:                             ____ ____
NEXT:                             sw   ____, ____
                                addi ____, ____, ____
                                jmp  ____
END:
```

ΜΕΡΟΣ Β

Υλοποιήστε την παρακάτω ρουτίνα σε assembly του MIPS. Δίνονται τα μεγέθη των τύπων char 8 bits, short int 16 bits και int 32 bits.

```

int foo(struct S *s) {
    int a, i;
    char b;
    short int c;
    char *d;

    a = 0xdeadbeef; b = 0; c = 0;
    d = s->D[1];

    for (i=0; i < 2; i++) a += s->A[i];
    for (i=0; i < 8; i++) b += s->B[i];
    for (i=0; i < 2; i++) c += s->C[i];

    if (a % 30 < 10)
        a = a / 30;

    s->B[0] = b;
    s->C[0] = c;
    return a;
}

```

ΜΕΡΟΣ Γ

Τα B+ δέντρα (B+-trees) είναι μία δομή δεδομένων που χρησιμοποιούνται ευρέως για την γρήγορη αναζήτηση σε συστήματα βάσεων δεδομένων¹. Παρακάτω σας δίνεται η δομή που αναπαριστά έναν κόμβο ενός B+ δέντρου καθώς και η αναδρομική συνάρτηση η οποία αναζητά ένα κλειδί (key) μέσα στο δέντρο και επιστρέφει 1 αν το κλειδί βρεθεί, αλλιώς 0. Υλοποιήστε τις ρουτίνες `get_children_index` και `bpt_lookup` σε assembly του MIPS.

```

struct bpt_node {
    int keys[4];
    struct bpt_node *children[5];
    int nr_keys;
    int is_leaf;
};

int get_children_index(int *A, int N, int key) {
    int i = 0;
    while (i < N && key < A[i]) i++;
    return i;
}

int bpt_lookup(struct bpt_node *n, int key) {
    int index, ret;
    if (n == 0) // 'n' is a NULL pointer
        return 0;

    index = get_children_index(&(n->keys[0]), n->nr_keys, key);
    if (n->is_leaf && n->keys[index] == key)
        ret = 1;
    else
        ret = bpt_lookup(n->children[index], key);

    return ret;
}

```

¹ https://en.wikipedia.org/wiki/B%2B_tree

Για την υλοποίηση της άσκησης μπορείτε να χρησιμοποιήσετε τον **MSIM**, ένα MIPS emulator που αναπτύχθηκε από συμφοιτητές σας και διατίθεται από το εργαστήριο Υπολογιστικών Συστημάτων (CSLab). Στον emulator αυτό, μπορείτε να γράφετε MIPS assembly και να την εκτελείτε παρακολουθώντας τα περιεχόμενα των καταχωρητών και της μνήμης καθιστώντας έτσι ευκολότερη την παραγωγή και τον έλεγχο του απαιτούμενου κώδικα. Τον MSIM μπορείτε να τον κατεβάσετε από τη σελίδα των ασκήσεων του site του μαθήματος.

Παραδοτέο της άσκησης θα είναι ένα ηλεκτρονικό κείμενο (**pdf, docx ή odt**) που θα περιέχει τους κώδικες assembly και των 3 μερών της άσκησης. Ο κώδικας θα πρέπει να περιέχει αναλυτικά σχόλια για την κατανόηση της λύσης σας από τους διδάσκοντες.

Στο ηλεκτρονικό κείμενο να αναφέρετε στην αρχή τα στοιχεία σας (Όνομα, Επώνυμο, ΑΜ).

Η άσκηση θα παραδοθεί ηλεκτρονικά στην ιστοσελίδα:

<http://www.cslab.ece.ntua.gr/courses/comparch/submit-tmima1>

Δουλέψτε ατομικά. Έχει ιδιαίτερη αξία για την κατανόηση του μαθήματος να κάνετε μόνοι σας την εργασία. Μην προσπαθήσετε να την αντιγράψετε από άλλους συμφοιτητές σας.