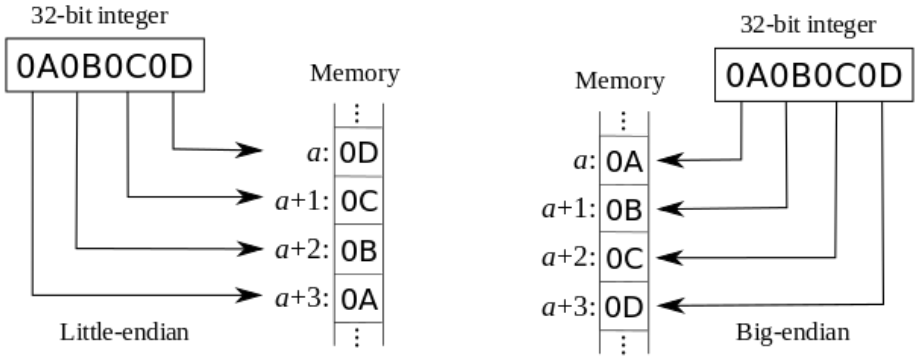
ΠΛΥ410 – ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΥΣΤΗΜΑΤΩΝ

Εργαστήριο #6
28/04/2015

I. BITWISE OPERATORS

Όπως γνωρίζετε από το μάθημα της Αρχιτεκτονικής Υπολογιστών, τα 4 bytes τα οποία καταλαμβάνει ένας 32-μπιτος αριθμός, μπορούν να αποθηκευτούν με δύο διαφορετικούς τρόπους. Ανάλογα με το ποιο από τα 4 bytes αποθηκεύεται πρώτο (στη μικρότερη διεύθυνση στη μνήμη), έχουμε είτε την αποθήκευση μικρού άκρου (little-endian) είτε την αποθήκευση μεγάλου άκρου (big endian), όπως φαίνεται στα παρακάτω σχήματα.



Επειδή δεν ακολουθούν όλα τα συστήματα την ίδια λύση, είναι πολλές φορές απαραίτητο να μετατρέπουμε τα δεδομένα μας από τη μία μορφή στην άλλη. Σας ζητείται να κάνετε τη μετατροπή αυτή χρησιμοποιώντας `bitwise operators` (&, |, «. »). Συγκεκριμένα, θα πρέπει να υλοποιήσετε μία συνάρτηση με πρωτότυπο:

```
int reverse_endian(int num);
```

η οποία θα επιστρέφει τον αριθμό όπου το 4ο byte έχει γίνει 1ο, το 3ο έχει γίνει 2ο, κλπ.

» *Μπορείτε να σκεφτείτε και έναν άλλο τρόπο, χωρίς τη χρήση bitwise operators;*

II. ΑΡΧΕΙΑ

Στο αρχείο `data.txt` υπάρχει μια σειρά από ακεραίους αριθμούς. Ο πρώτος από αυτούς τους αριθμούς καθορίζει πόσοι ακόμα αριθμοί υπάρχουν στο αρχείο. Αν για παράδειγμα αυτός ο αριθμός είναι ο 4, τότε ακολουθούν άλλοι 4 ακέραιοι αριθμοί στο αρχείο.

α) Να γραφεί πρόγραμμα το οποίο ανοίγει αυτό το αρχείο, διαβάζει τα δεδομένα και τα αποθηκεύει σε έναν κατάλληλο πίνακα ακεραίων που θα δημιουργηθεί δυναμικά με την χρήση της `malloc()`. Στην συνέχεια το πρόγραμμα θα πρέπει να αποθηκεύσει τα δεδομένα που διάβασε σε δύο διαφορετικά αρχεία: το αρχείο `positive.txt` και το αρχείο `negative.txt`. Στο αρχείο `positive.txt` θα αποθηκευτούν οι θετικοί αριθμοί από αυτούς που διάβασε και στο αρχείο `negative.txt` θα αποθηκευτούν οι αρνητικοί αριθμοί. Τόσο το αρχείο `positive.txt` όσο και το αρχείο `negative.txt` να διατηρήσουν τη μορφοποίηση του αρχικού αρχείου, δηλαδή στην πρώτη γραμμή κάθε αρχείου θα πρέπει να υπάρχει το πλήθος των αριθμών που ακολουθούν.

β) Να προστεθεί κώδικας ο οποίος επαναφέρει τη θέση του αρχείου `positive.txt` στην αρχή, ξαναδια-

βάζει τους αριθμούς από το αρχείο `positive.txt` και υπολογίζει το άθροισμά τους.

III. VARIADIC FUNCTIONS

Σε αυτή την εργασία θα φτιάξετε μία συνάρτηση variadic η οποία όπως έχουμε πει στο μάθημα, είναι συνάρτηση με μεταβλητό πλήθος παραμέτρων. Συγκεκριμένα, υλοποιήσετε μία συνάρτηση `myprint()` η οποία παίρνει ως πρώτο όρισμα το πλήθος των τιμών, τις οποίες πρέπει να εκτυπώσει. Οι τιμές μπορεί να είναι είτε ακέραιες είτε πραγματικές. Οπότε πριν από κάθε τιμή θα υπάρχει μία επιπλέον παράμετρος που θα δείχνει τον τύπο της τιμής που ακολουθεί, και συγκεκριμένα, ο χαρακτήρας 'd' για ακέραιο και 'f' για float. Έτσι για παράδειγμα η ακόλουθη εντολή:

```
myprint(3, 'd', 12, 'f', 23.4, 'd', 14)
```

θα εκτυπώσει 3 τιμές (ενώ συνολικά δέχεται 7 ορίσματα). Η πρώτη και η τρίτη τιμή είναι ακέραιες ενώ η δεύτερη πραγματική.

IV. ΕΠΙΠΛΕΟΝ ΕΞΑΣΚΗΣΗ ΜΕ BITWISE OPERATORS

Να γράψετε μία συνάρτηση `void printbin(int n)` η οποία τυπώνει τον αριθμό n στο δυαδικό σύστημα (δηλαδή τυπώνει τα bits του αριθμού). Να χρησιμοποιήσετε `bitwise operators` (&, |, «. »,)˜ώστε να βρίσκονται ένα-ένα τα 32 bits μέσω του αλγορίθμου που περιγράφεται παρακάτω. Για δοκιμή της `printbin()`, η `main()` θα την καλεί για όσους αριθμούς θέλετε και πάντως τουλάχιστον για το 0, το 1 και το -1.

Αλγόριθμος Εκτύπωσης Bits

```
Δεδομένα: Αριθμός για εκτύπωση ( n )

1. Δημιουργία μάσκας για έλεγχο του πιο σημαντικού bit:
     mask = 1, ολισθημένο κατά 31 θέσεις αριστερά

2. Εκτύπωση των bits του n:
     Επανάληψη 32 φορές:
        Αν (χρησιμοποιώντας το mask) το πιο σημαντικό bit είναι 0 τότε
           εκτύπωσε 0
        αλλιώς
           εκτύπωσε 1
        Τέλος-Αν
        Ολίσθησε το n μια θέση αριστερά
     Τέλος-Επανάληψη
```

V. ΕΠΙΠΛΕΟΝ ΕΞΑΣΚΗΣΗ ΜΕ ΑΡΧΕΙΑ

Για επιπλέον εξάσκηση με αρχεία, μπορείτε να επαναλάβετε την εργαστηριακή άσκηση 5(I) όπου στο τέλος, τον πίνακα από δομές τον αποθηκεύετε σε ένα αρχείο κειμένου (αποθηκεύετε ένα-ένα τα πεδία, σε ξεχωριστές γραμμές, με fprintf()).