ΜΥΥ502 – ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΥΣΤΗΜΑΤΩΝ

Εργαστήριο #7
29/11/2016

I. ΑΝΤΙΓΡΑΦΗ ΑΡΧΕΙΟΥ (MINI CP)

Σε αυτή την εργασία καλείστε να επεξεργαστείτε δυαδικά αρχεία με τις συναρτήσεις `open()`, `read()`, `write()`, `close()`, υλοποιώντας μία απλή εκδοχή της εντολής τερματικού «`cp`».

Πρέπει να φτιάξετε ένα πρόγραμμα το οποίο δέχεται ως ορίσματα στη `main()` τα ονόματα δύο (δυαδικών) αρχείων και αντιγράφει το πρώτο στο δεύτερο. Συγκεκριμένα, θα ανοίγει το πρώτο δυαδικό αρχείο για διάβασμα και το δεύτερο για δημιουργία (με μηδενισμό) και γράψιμο—δώστε αρχικές άδειες τύπου `0755` στο νέο αρχείο. Μέσα σε ένα βρόχο, θα διαβάζει 1 byte τη φορά από το πρώτο και θα το γράφει στο δεύτερο αρχείο.

Δοκιμάστε να αντιγράψετε το `a.out` σε ένα άλλο αρχείο (π.χ. `new.out`), το οποίο προφανώς πρέπει να το κάνετε εκτελέσιμο. Αν όλα πάνε καλά θα πρέπει να μπορείτε να τρέξετε το νέο αρχείο όπως έτρεχε το `a.out`.

II. ΑΡΧΕΙΑ ΒΙΝΤΕΟ AVI

Στην άσκηση αυτή θα δημιουργήσετε ένα πρόγραμμα το οποίο βρίσκει και εμφανίζει βασικά στοιχεία για ένα αρχείο video τύπο AVI. Όλα τα αρχεία βίντεο αυτού του τύπου στα αρχικά bytes τους περιέχουν ορισμένες πληροφορίες ('μετα-πληροφορίες') σχετικά με το βίντεο που αποθηκεύουν. Αυτό το τμήμα του αρχείου καλείται *κεφαλίδα* (header). Στον Πίνακα 1 φαίνεται η σειρά και σε ποια θέση είναι αποθηκευμένες οι πληροφορίες της κεφαλίδας όλων των αρχείων AVI. Να σημειωθεί ότι **η κεφαλίδα ξεκινά μετά τα πρώτα 32 bytes από την αρχή του αρχείου** και έχει συνολικό μέγεθος 56 bytes. Έτσι το αρχικό byte της κεφαλίδας είναι στο 32ο byte του αρχείου.

Ζητείται να φτιάξετε ένα πρόγραμμα το οποίο θα ανοίγει ένα αρχείο AVI, θα διαβάζει ορισμένες πληροφορίες από την κεφαλίδα του και θα τις εμφανίζει στην οθόνη. Συγκεκριμένα θα πρέπει να φτιάξετε μια `main` η οποία με χρήση ορισμάτων (argc, argv) θα δέχεται το όνομα ένος αρχείου AVI το οποίο και θα ανοίγει (ως δυαδικό) με την `open()`. Στη συνέχεια, κάνοντας χρήση των συναρτήσεων `read()` και `lseek()` θα διαβάζει τους κατάλληλους ακεραίους ώστε να υπολογίζει και να εμφανίζει τις εξής πληροφορίες:

1. Μέγεθος εικόνας σε pixel (πλάτος επί ύψος)
2. Frame rate (καρέ ανά δευτερόλεπτο, fps)
3. Διάρκεια βίντεο (σε δευτερόλεπτα)

Για το 1, η πληροφορία είναι έτοιμη στην κεφαλίδα (width/height of video image in pixels). Για το 2, θα πρέπει να υπολογίσετε:

$$\text{καρέ ανά δευτερόλεπτο (fps)} = \text{data\_rate}/\text{time\_scale},$$

τα οποία δίνονται επίσης προς το τέλος της κεφαλίδας. Τέλος για το 3, από τη στιγμή που έχετε τα καρέ ανά

Πίνακας 1: *Κεφαλίδα AVI. Τα μεγέθη είναι σε bytes. Θυμηθείτε ότι η κεφαλίδα ξεκινά στο 32ο byte του αρχείου.*

offset	size	description
0	4	time delay between frames in microseconds
4	4	data rate of AVI data
8	4	padding multiple size, typically 2048
12	4	parameter flags
16	4	number of video frames
20	4	number of preview frames
24	4	number of data streams (1 or 2)
28	4	suggested playback buffer size in bytes
32	4	width of video image in pixels
36	4	height of video image in pixels
40	4	time scale, typically 30
44	4	data rate (frame rate = data rate / time scale)
48	4	starting time, typically 0

δευτερόλεπτο, η διάρκεια προκύπτει από τον συνολικό αριθμό καρέ:

$$\text{διάρκεια (sec)} = \text{number\_of\_video\_frames/fps}.$$

III. BITWISE OPERATORS: ENDIANESS

Θυμηθείτε την άσκηση VI του Εργαστηρίου #2. Επειδή δεν ακολουθούν όλα τα συστήματα την ίδια λύση, είναι πολλές φορές απαραίτητο να μετατρέπουμε τα δεδομένα μας από τη μία μορφή endianess στην άλλη. Σας ζητείται να κάνετε τη μετατροπή αυτή χρησιμοποιώντας **bitwise operators** (`&,|,<<,>>`) — στη συνέχεια αν θέλετε μπορείτε να κάνετε και δεύτερη λύση χωρίς 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.

IV. VARIADIC FUNCTIONS

Σε αυτή την εργασία θα φτιάξετε μία συνάρτηση με μεταβλητό πλήθος παραμέτρων. Συγκεκριμένα, θα πρέπει να υλοποιήσετε τη συνάρτηση `printnums()` 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 'i' για ακέραιο και 'd' για double. Έτσι για παράδειγμα η ακόλουθη εντολή:

```
printnums(3, 'i', 12, 'd', 23.4, 'i', 14)
```

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

V. ΕΠΙΠΛΕΟΝ ΕΞΑΣΚΗΣΗ: 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.

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
```

VI. ΕΠΙΠΛΕΟΝ ΕΞΑΣΚΗΣΗ: ΔΥΑΔΙΚΑ ΑΡΧΕΙΑ

Αναζητείστε πληροφορίες για δυαδικά αρχεία που σας φαίνονται ενδιαφέροντα, π.χ.

- αρχεία μουσικής (mp3, wav, κλπ)
- αρχεία βίντεο (avi, 3gp, κλπ)
- αρχεία συστήματος (π.χ. η δομή ELF του αρχείου a.out)

Αφού βρείτε της πληροφορίες για το πώς είναι οργανωμένα τα αρχεία και οι κεφαλίδες τους, προσπαθήστε να φτιάξετε προγράμματα που τα ανοίγουν και εκτυπώνουν χρήσιμες πληροφορίες για αυτά. Τέλος, **ενημερώστε τον διδάσκοντα ώστε να αποτελέσουν πιθανές εργασίες εργαστηρίου στο μέλλον** :-)