Εργαστήριο 10: Είσοδος και Έξοδος με Αρχεία

Στο εργαστήριο αυτό θα μελετήσουμε τους μηχανισμούς εισόδου/εξόδου που μας παρέχει η C. Θα αναφερθούμε στις μονάδες εισόδου/εξόδου, που είναι τα ρεύματα, θα κάνουμε μία επισκόπηση στα προκαθορισμένα ρεύματα και θα ορίσουμε δικά μας ρεύματα για την επεξεργασία αρχείων κειμένου και δυαδικών αρχείων.

Άσκηση 1: Αρχεία κειμένου - (more.c)

Κατασκευάστε το πρόγραμμα `more.c` που δέχεται ως όρισμα γραμμής εντολής το όνομα ενός αρχείου κειμένου και προβάλλει ανά 20 τις γραμμές του αρχείου, προτρέποντας τον χρήστη να συνεχίσει, αν επιθυμεί, έως ότου συναντήσει το τέλος του αρχείου.

Συναρτήσεις Εισόδου/Εξόδου στην C

Ανοιγμα και Κλείσιμο Αρχείων

```
FILE *fopen(const char *filename, const char *mode);
int fclose(FILE *fp);
```

- `fopen`: Ανοίγει το αρχείο με όνομα `filename` με τρόπο προσπέλασης που καθορίζεται από το `mode`. Επιστρέφει ένα ρεύμα μέσω του οποίου μπορούμε στη συνέχεια να αναφερόμαστε στο αρχείο, ή `NULL` αν δεν ήταν δυνατόν να ανοίξει το αρχείο.
  - `mode` **Παραδείγματα**:
    - `"r"`: Διάβασμα από υπάρχον αρχείο.
    - `"w"`: Γράψιμο σε αρχείο. Δημιουργείται αν δεν υπάρχει ή διαγράφονται τα περιεχόμενα αν υπάρχει.
    - `"a"`: Προσάρτηση δεδομένων στο τέλος του αρχείου.
    - `"r+"`, `"w+"`, `"a+"`: Διάφοροι συνδυασμοί ανάγνωσης και εγγραφής.
- `fclose`: Κλείνει το ρεύμα `fp`. Επιστρέφει `0` σε περίπτωση επιτυχίας.

Άλλες Συναρτήσεις

1. `int feof(FILE *fp)`
   - Επιστρέφει τιμή διαφορετική από το 0 αν το προηγούμενο διάβασμα απέτυχε λόγω τέλους αρχείου, αλλιώς επιστρέφει 0.
2. `int fprintf(FILE *fp, ...)`
   - Αντίστοιχη της `printf`, αλλά γράφει στο ρεύμα `fp` αντί του `stdout`.
3. `int fscanf(FILE *fp, ...)`
   - Αντίστοιχη της `scanf`, αλλά διαβάζει από το ρεύμα `fp` αντί του `stdin`.
4. `char *fgets(char *buf, int max, FILE *fp)`
   - Διαβάζει το πολύ `max-1` χαρακτήρες από το ρεύμα `fp` μέχρι την αλλαγή γραμμής και τους φυλάσσει στο `buf`. Αν δεν υπάρχουν δεδομένα, επιστρέφει `NULL`.
5. `int getc(FILE *fp)`

- Επιστρέφει τον επόμενο χαρακτήρα από το ρεύμα `fp`, ή `EOF` αν φτάσει στο τέλος του αρχείου.

Άσκηση 2: Δυαδικά αρχεία - (grades.c)

2.1 Κατασκευάστε το πρόγραμμα `grades.c`, το οποίο να ανοίγει το αρχείο `grades.dat` για γράψιμο σε δυαδική μορφή. Στη συνέχεια, να διαβάζει ένα όνομα (συμβολοσειρά) από την πρότυπη είσοδο και έναν βαθμό και να τα γράφει στο αρχείο. Το πρόγραμμα να τερματίζει όταν επισημανθεί, με κάποιο τρόπο, το τέλος της εισόδου.

2.2 Στη συνέχεια, επεκτείνετε το πρόγραμμά σας, ώστε να ανοίγει το αρχείο `grades.dat` για διάβασμα, να διαβάζει τα δεδομένα που έχουν γραφεί σ' αυτό και να τα προβάλλει στην οθόνη.

Συναρτήσεις για Δυαδική Εισαγωγή/Εξαγωγή:

1. `size_t fread(void *ptr, size_t size, size_t count, FILE *fp)`
   - Διαβάζει από το ρεύμα `fp`, το πολύ `count` δεδομένα μεγέθους `size` το καθένα, από τη διεύθυνση `ptr`.
2. `size_t fwrite(const void *ptr, size_t size, size_t count, FILE *fp)`
   - Γράφει στο ρεύμα `fp` το πολύ `count` δεδομένα μεγέθους `size` το καθένα, από τη διεύθυνση `ptr`.

Για να δείτε τα περιεχόμενα του αρχείου `grades.dat`, αν δουλεύετε σ' ένα Unix σύστημα, χρησιμοποιήστε την εντολή `od -tu1c grades.dat` ή `hexdump -C grades.dat`.

Άσκηση 3: Σύγκριση Αρχείων - (filediff.c)

Δημιουργήστε το πρόγραμμα compare.c, το οποίο να δέχεται στη γραμμή εντολής τα ονόματα δύο αρχείων και να ελέγχει αν τα αρχεία αυτά είναι ίδια byte προς byte.

Άσκηση 4: Μέτρηση Στατιστικών Αρχείων - (count.c)

Στην άσκηση αυτή θα υλοποιήσουμε ένα υποσύνολο της εντολής wc του Unix. Συγκεκριμένα, θα κατασκευάσουμε πρόγραμμα που θα μετράει το πλήθος των χαρακτήρων και το πλήθος των γραμμών ενός αρχείου κειμένου.

4.1 Κατασκευάστε το πρόγραμμα `count.c` που να δέχεται ως όρισμα γραμμής εντολής το όνομα ενός αρχείου κειμένου, να το ανοίγει για διάβασμα, να μετράει το πλήθος των χαρακτήρων του αρχείου και να προβάλλει το αποτέλεσμα στην οθόνη.

4.2 Τροποποιήστε το πρόγραμμά σας, ώστε να μετράει και το πλήθος των γραμμών του αρχείου.