

ΕΡΓΑΣΤΗΡΙΟ 11: Είσοδος/Εξόδος - Αρχαία

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: Αρχαία κειμένου

Κατασκευάστε το πρόγραμμα `display.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.

<pre>FILE *fopen(const char *filename, const char *mode)</pre> <p>Ανοίγει το αρχείο με όνομα <code>filename</code> με τρόπο προσπέλασης που καθορίζεται από το <code>mode</code>:</p> <ul style="list-style-type: none">◦ <code>"r"</code>: Διάβασμα από υπάρχον αρχείο.◦ <code>"w"</code>: Γράψιμο σε αρχείο. Αν το αρχείο δεν υπάρχει, δημιουργείται. Αν υπάρχει, διαγράφονται τα περιεχόμενά του και το γράψιμο αρχίζει από την αρχή.◦ <code>"a"</code>: Προσάρτηση δεδομένων στο τέλος του αρχείου, χωρίς διαγραφή των υπαρχόντων περιεχομένων του.◦ <code>"r+"</code>: Διάβασμα και γράψιμο οπουδήποτε στο αρχείο χωρίς διαγραφή των υπαρχόντων περιεχομένων του.◦ <code>"w+"</code>: Διάβασμα και γράψιμο οπουδήποτε στο αρχείο με διαγραφή των υπαρχόντων περιεχομένων του.◦ <code>"a+"</code>: Διάβασμα οπουδήποτε μέσα από το αρχείο και γράψιμο μόνο στο τέλος του, χωρίς διαγραφή των υπαρχόντων περιεχομένων του.◦ Στις παραπάνω συμβολοσειρές που δείχνουν τον τρόπο προσπέλασης του αρχείου προστίθεται και ο χαρακτήρας <code>b</code>, αν πρόκειται για δυαδικό αρχείο (<code>"rb"</code>, <code>"wb"</code>, <code>"ab"</code>, <code>"rb+"</code> ή <code>"r+b"</code>, <code>"wb+"</code> ή <code>"w+b"</code>, <code>"ab+"</code> ή <code>"a+b"</code>). <p>Επιστρέφει ένα ρεύμα μέσω του οποίου μπορούμε στη συνέχεια να αναφερόμαστε στο αρχείο, ή <code>NULL</code>, αν για κάποιο λόγο δεν ήταν δυνατόν να ανοίξει το αρχείο.</p>	<pre>int fclose(FILE *fp)</pre> <p>Κλείνει το ρεύμα <code>fp</code>. Επιστρέφει 0 σε περίπτωση επιτυχίας.</p> <pre>int feof(FILE *fp)</pre> <p>Επιστρέφει τιμή διάφορη από το 0, αν προηγούμενο διάβασμα απέτυχε λόγω του ότι τα δεδομένα στο αρχείο έχουν τελειώσει, αλλιώς επιστρέφει 0.</p> <pre>int fprintf(FILE *fp, ...) int fscanf(FILE *fp, ...)</pre> <p>Ίδιες με τις <code>printf</code> και <code>scanf</code>, μόνο που γράφουν στο ρεύμα <code>fp</code> αντί του <code>stdout</code> ή διαβάζουν από το ρεύμα <code>fp</code> αντί του <code>stdin</code>, αντίστοιχα.</p> <pre>char *fgets(char *buf, int max, FILE *fp)</pre> <p>Διαβάζει το πολύ <code>max-1</code> χαρακτήρες από το ρεύμα <code>fp</code> μέχρι την αλλαγή γραμμής και τους φυλάσσει στο <code>buf</code> (συμπεριλαμβανομένης και της αλλαγής γραμμής). Αν δεν υπάρχουν δεδομένα για διάβασμα από το ρεύμα, επιστρέφει <code>NULL</code>.</p> <pre>int getc(FILE *fp)</pre> <p>Επιστρέφει τον επόμενο χαρακτήρα από το ρεύμα <code>fp</code>, η <code>EOF</code>, αν έχουμε φτάσει στο τέλος του αρχείου.</p>
---	---

Άσκηση 2: Δυαδικά αρχεία

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.

```
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 και τα τοποθετεί από τη διεύθυνση `ptr` και μετά.

```
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, παίρνοντάς τα από τη διεύθυνση `ptr` και μετά.

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

Άσκηση 3: 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: 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.