

Εργασία 3 – Βάση ψηφιακών αντικειμένων σε δυαδικό αρχείο

Αναπτύξτε ένα πρόγραμμα που υλοποιεί μια βάση “ψηφιακών αντικειμένων” μέσω ενός δυαδικού αρχείου (binary file). Ψηφιακό αντικείμενο είναι οτιδήποτε μπορεί να αναπαρασταθεί ως μια σειρά από bytes, ενώ η βάση μπορεί να περιέχει πολλά τέτοια αντικείμενα.

Η επιθυμητή λειτουργικότητα περιγράφεται παρακάτω. Η ακριβής μορφή των μηνυμάτων που εκτυπώνονται από το πρόγραμμα περιγράφεται σε ξεχωριστή ενότητα.

Περιγραφή λειτουργίας

Το πρόγραμμα δέχεται και εκτελεί τις εξής εντολές σε επανάληψη:

- **o(open) <dbname>**: Άνοιγμα κι έλεγχος αρχείου βάσης. Ο χρήστης δίνει το όνομα του αρχείου dbname στο οποίο (θα) είναι αποθηκευμένη η βάση. Αν έχει ήδη ανοιχτεί κάποιο αρχείο βάσης, αυτό κλείνει πριν ανοιχτεί το νέο. Αν το αρχείο δεν υπάρχει, δημιουργείται και αρχικοποιείται κατάλληλα. Αν υπάρχει αλλά δεν έχει κατάλληλη δομή (δεν υπάρχει η αναμενόμενη μετα-πληροφορία τύπου - βλ. παρακάτω), τότε εκτυπώνεται μήνυμα λάθους. Αν το αρχείο δε μπορεί να ανοιχτεί/δημιουργηθεί, τότε εκτυπώνεται μήνυμα λάθους.
- **i(mport) <fname> <objname>**: Εισαγωγή αντικειμένου. Ο χρήστης δίνει το όνομα του αρχείου fname που περιέχει το αντικείμενο που θέλει να εισάγει στη βάση, και το όνομα objname με το οποίο επιθυμεί να αποθηκεύσει το αντικείμενο στην βάση. Αν το αρχείο fname που έδωσε ο χρήστης δεν υπάρχει, εκτυπώνεται μήνυμα λάθους. Αν υπάρχει ήδη στη βάση αντικείμενο με το ίδιο όνομα name, τότε εκτυπώνεται μήνυμα λάθους.
- **f(ind) <objname>**: Αναζήτηση αντικειμένων. Ο χρήστης δίνει το όνομα objname του αντικειμένου που αναζητά. Το πρόγραμμα ψάχνει για όλα τα αντικείμενα στη βάση με ονόματα που περιέχουν (σε οποιοδήποτε σημείο τους) το όνομα που έδωσε ο χρήστης, και εκτυπώνει τα πλήρη ονόματα τους. Εάν ο χρήστης εισάγει "*" τότε το πρόγραμμα εκτυπώνει τα ονόματα όλων των αντικειμένων που υπάρχουν στην βάση.
- **e(xport) <objname> <fname>**: Εξαγωγή αντικειμένου. Ο χρήστης δίνει το όνομα του αντικειμένου objname προς εξαγωγή και το όνομα του αρχείου fname στο οποίο θέλει να το αποθηκεύσει. Το πρόγραμμα εντοπίζει το αντικείμενο στην βάση και αντιγράφει τα περιεχόμενά του στο αρχείο που υπέδειξε ο χρήστης. Αν το αντικείμενο δεν υπάρχει στην βάση, εκτυπώνεται μήνυμα λάθους. Αν το αρχείο που υπέδειξε ο χρήστης δεν υπάρχει, δημιουργείται από το πρόγραμμα. Αν το αρχείο δε μπορεί να δημιουργηθεί ή υπάρχει ήδη, εκτυπώνεται μήνυμα λάθους. Σε κάθε περίπτωση, το αντικείμενο παραμένει στην βάση.
- **d(etele) <objname>**: Διαγραφή αντικειμένου. Ο χρήστης δίνει το όνομα του αντικειμένου που θέλει να διαγράψει από τη βάση. Το πρόγραμμα εντοπίζει το αντικείμενο, το διαγράφει, και το αρχείο της βάσης κόβεται κατάλληλα. Αν το αντικείμενο δεν υπάρχει, εκτυπώνεται μήνυμα λάθους.
- **c(lose)**: Κλείσιμο αρχείου βάσης.
- **q(uit)**: Τερματισμός. Το πρόγραμμα τερματίζει, αφού πρώτα κλείσει το αρχείο της βάσης (αν ο χρήστης δεν έχει ήδη κλείσει το αρχείο της βάσης μέσω της close).

Είσοδος/έξοδος προγράμματος

Πριν και μετά από κάθε μήνυμα εξόδου εκτυπώνεται χαρακτήρας αλλαγής γραμμής.

| Λειτουργία | Είσοδος | Έξοδος stdout (επιτυχία) | Έξοδος stdout (αποτυχία) |
|------------|------------------|---|---|
| open | o <dbname> | | Error opening <dbname>. ή Invalid db file <dbname>. |
| import | i <fname> <name> | | File <fname> not found. ή Object <name> already in db. ή No open db file. |
| find | f <name> | ## <object name> ... <object name> | No open db file. |
| export | e <name> <fname> | | Object <name> not in db. ή Cannot open file <fname>. ή No open db file. |
| delete | d <name> | | Object <name> not in db. ή No open db file. |
| close | c | | No open db file. |
| quit | q | | |

Απαιτήσεις υλοποίησης

- Ορίστε την διεπαφή των λειτουργιών της βάσης μέσω ενός header file (**objdb.h**) που περιέχει τους ορισμούς για τύπους δεδομένων που πιθανώς έχετε ορίσει καθώς και τα πρωτότυπα των συναρτήσεων (function prototypes) για τις λειτουργίες της βάσης (αρχικοποίηση νέου αρχείου βάσης με προσθήκη πληροφορίας τύπου, έλεγχος πληροφορίας τύπου για υπάρχον αρχείο βάσης, αναζήτηση, εισαγωγή, εξαγωγή,, διαγραφή αντικειμένων, κλείσιμο αρχείου βάσης).

- Σε ξεχωριστό αρχείο (**objdb.c**), υλοποιήστε τις λειτουργίες που ορίζονται στο header file. Οι συναρτήσεις δεν πρέπει να διαβάζουν από τη συμβατική είσοδο, ούτε να εκτυπώνουν στη συμβατική έξοδο. Ειδικά για τη συνάρτηση αναζήτησης, σκεφτείτε με ποιο τρόπο θα επιστρέφει τα αποτελέσματα στο κυρίως πρόγραμμα έτσι ώστε αυτό να τα εκτυπώσει όποτε αυτό χρειάζεται. Οι συναρτήσεις που υλοποιούν τις λειτουργίες εισαγωγής, εξαγωγής και διαγραφής πρέπει να χρησιμοποιούν (εσωτερικά) τη συνάρτηση αναζήτησης. Κατά την εισαγωγή ενός αντικειμένου στη βάση, το μέγεθος του αντικειμένου πρέπει να αποθηκευτεί ως int (binary) και όχι ως string.
- Αναπτύξτε ένα πρόγραμμα ελέγχου (**project3.c**), που επιτρέπει στον χρήστη να δημιουργεί νέες ή να ανοίγει υπάρχουσες βάσεις και να πραγματοποιεί σε αυτές τις παραπάνω λειτουργίες δίνοντας κατάλληλες εντολές.
- Γράψτε κατάλληλο **Makefile**. Τα δύο .c αρχεία πρέπει να μεταγλωττίζονται χωριστά. Το τελικό εκτελέσιμο πρέπει να λέγεται project3 και να παράγεται από τον πρώτο στόχο.
- Μπορείτε να υποθέσετε ότι το μέγιστο μέγεθος των ονομάτων dbname, fname και name που δίνει ο χρήστης είναι 255 bytes (χωρίς τον τερματικό χαρακτήρα). Τα ονόματα dbname, fname μπορεί να περιέχουν και πληροφορία μονοπατιού.
- Η πρόσβαση στα αρχεία πρέπει να γίνει χρησιμοποιώντας τις λειτουργίες της βιβλιοθήκης stdio. Το πρόγραμμά σας πρέπει να ελέγχει την τιμή επιστροφής κάθε κλήσης (καθώς και μέσω της λειτουργίας ferror) για το αν η ζητούμενη λειτουργία εκτελέστηκε επιτυχώς. Αν υπάρξει σφάλμα που δεν μπορεί να διαχειριστεί το ίδιο το πρόγραμμα έτσι ώστε να μπορεί να συνεχιστεί απρόσκοπτα η εκτέλεση του, το πρόγραμμα πρέπει να τερματίζει, αφού εκτυπώσει κατάλληλο μήνυμα λάθους στο stderr.
- Πρέπει να εισάγεται σε επιλεγμένα σημεία στο αρχείο της βάσης μετα-πληροφορία με βάση την οποία θα μπορεί να γίνεται έλεγχος κατά πόσο το αρχείο έχει την αναμενόμενη/σωστή μορφή/δομή (περισσότερα γι αυτό, στο φροντιστήριο).
- Η αναζήτηση των αντικειμένων που βρίσκονται στην βάση πρέπει να γίνεται διαβάζοντας κάθε φορά τα περιεχόμενα του αρχείου της βάσης. Δεν επιτρέπεται να αποθηκεύονται πληροφορίες "ευρετηρίου" στη μνήμη του προγράμματος.
- Η μεταφορά των περιεχομένων των αντικειμένων από/προς την βάση πρέπει να γίνεται τμηματικά μέσω ενός πίνακα μεγέθους 512 bytes. Η μνήμη γι αυτό μπορεί να δεσμεύεται από το πρόγραμμα στατικά ή δυναμικά (ότι νομίζετε καλύτερο).
- Το κόψιμο του αρχείου της βάσης κατά τη διαγραφή αντικειμένων πρέπει, κατ' εξαίρεση, να γίνει χρησιμοποιώντας τη λειτουργία συστήματος truncate (θυμίζουμε ότι η βιβλιοθήκη stdio δεν παρέχει λειτουργία για την μείωση του μεγέθους αρχείων). **Προσοχή:** Πριν καλέσετε την truncate πρέπει προηγουμένως να κλείσετε το αρχείο της βάσης με fclose. Μετά την κλήση της truncate μπορείτε να ανοίξετε πάλι το αρχείο με fopen.
- Ανά πάσα στιγμή πρέπει να υπάρχει το πολύ μια ανοιχτή ροή στο αρχείο της βάσης. Με άλλα λόγια, δεν επιτρέπεται να καλέσετε την fopen πολλές/διαδοχικές φορές για το αρχείο

της βάσης, όμως επιτρέπεται να καλέσετε την `open` για να δημιουργήσετε μια νέα ροή στο αρχείο της βάσης αφού κλείσετε την προηγούμενη ροή με `fclose`.

- Το μήνυμα “No open db file.” εκτυπώνεται αν ο χρήστης προσπαθήσει να εκτελέσει μία από τις σχετικές λειτουργίες χωρίς πρώτα να έχει ανοιχτεί επιτυχώς μια βάση.
- Απαγορεύεται η χρήση βοηθητικού αρχείου για την υλοποίηση λειτουργιών της βάσης.
- Απαγορεύονται οι καθολικές/static μεταβλητές και η χρήση `goto` και `gets`.
- Δώστε περιγραφικά ονόματα σε συναρτήσεις και μεταβλητές, σχολιάστε τις συναρτήσεις και όσα τμήματα του κώδικα δεν είναι ξεκάθαρα, και γράψτε ευανάγνωστα με σωστή στοίχιση. Κώδικας που “δεν διαβάζεται”, απορρίπτεται χωρίς εξέταση.

Συστάσεις για την ανάπτυξη του κώδικα

1. Σχεδιάστε την οργάνωση των δεδομένων στο αρχείο της βάσης, έτσι ώστε να υποστηρίζονται με σχετική ευκολία οι διαδικασίες εισαγωγής, αναζήτησης, εξαγωγής και διαγραφής. Χρησιμοποιήστε χαρτί και μολύβι για να σχεδιάσετε την οργάνωση που έχετε στο μυαλό σας. Μην αρχίσετε να γράφετε κώδικα αν δεν βεβαιωθείτε ότι το σχήμα οργάνωσης του αρχείου υποστηρίζει όλες τις ανάγκες του προγράμματος.
2. Αναπτύξτε τις βασικές συναρτήσεις στο **objdb.c** και τις αντίστοιχες λειτουργίες του προγράμματος, με δομημένο τρόπο και με την εξής σειρά: (α) αρχικοποίηση νέου αρχείου βάσης με πληροφορία τύπου, (β) έλεγχος πληροφορίας τύπου σε υπάρχον αρχείο βάσης, (γ) εισαγωγή, (δ) αναζήτηση, (ε) εξαγωγή, (στ) διαγραφή. Κάθε φορά που ολοκληρώνετε την υλοποίηση μιας νέας λειτουργίας, κάντε αντίστοιχες δοκιμές (μέσα από το κυρίως πρόγραμμα που θα καλεί την λειτουργία όταν το ζητήσει ο χρήστης) για να βεβαιωθείτε για την ορθότητα της υλοποίησης, προτού προχωρήσετε στην υλοποίηση της επόμενης συνάρτησης/λειτουργίας του προγράμματος. Κάντε δοκιμές με «οριακές» περιπτώσεις, π.χ., εισαγωγή σε κενή βάση, αναζήτηση σε κενή βάση, αναζήτηση αντικειμένου που δεν υπάρχει στην βάση, αναζήτηση αντικειμένου στην αρχή/τέλος του αρχείου, διαγραφή από την αρχή /τέλος της βάσης, κλπ.
3. Εκτός από τις βασικές λειτουργίες που θα υλοποιήσετε στο `objdb.c` για τις οποίες υποχρεωτικά καλείστε να φτιάξετε ξεχωριστές συναρτήσεις, προσπαθήστε να δημιουργήσετε συναρτήσεις για όλα τα τμήματα του κώδικα που επαναλαμβάνονται σε περισσότερα σημεία, καθώς και για επιμέρους λειτουργίες που θέλετε να ελέγξετε αυτόνομα.

Παρατηρήσεις

- Δώστε ιδιαίτερη προσοχή στην απαίτηση για το πόσες ανοιχτές ροές πρέπει να υπάρχουν ανά πάσα στιγμή στο αρχείο.
- Βεβαιωθείτε ότι το πρόγραμμά σας δεν προσπαθεί να εκτελέσει λειτουργίες `import`, `export` κτλ. σε βάση που δεν έχει ανοιχτεί.
- Αν το `autolab` σας παρουσιάζει μήνυμα λάθους “UnicodeDecodeError: 'utf-8' codec can't decode byte” τότε το πρόγραμμά σας εκτυπώνει στη συμβατική έξοδο δεδομένα που δεν

είναι `ascii` (μάλλον κατά τη διαδικασία της `find` και μάλλον γιατί έχετε κάνει λάθος στο “μέτρημα” των `bytes`).

- **ΕΛΕΓΧΕΤΕ ΠΑΝΤΑ ΤΙ ΣΑΣ ΕΠΙΣΤΡΕΦΟΥΝ** οι `fread`, `fseek`, κτλ.

Για όσους έχουν κέφι

- Επεκτείνετε την υλοποίησή σας έτσι ώστε να μπορεί να γίνει εισαγωγή και εξαγωγή ολόκληρων υποκαταλόγων (όχι μόνο μεμονωμένων αρχείων). Συγκεκριμένα, αν το όνομα που δίνεται στην `import` είναι ένας κατάλογος, τότε θα πρέπει στην βάση να εισάγονται, κατάλληλα συσχετισμένα με το όνομα του καταλόγου, όλα τα (κανονικά) αρχεία που περιέχει ο κατάλογος (χωρίς τυχόν ειδικά αρχεία ή άλλους υποκαταλόγους). Αντίστοιχα, αν το όνομα που δίνεται στην `export/delete` είναι το όνομα ενός καταλόγου τότε θα πρέπει να εξαγεται στον τοπικό κατάλογο / να σβήνεται από τη βάση ο κατάλογος με όλα τα αρχεία που ανήκουν σε αυτόν. Τέλος, η `find` θα πρέπει να λειτουργεί και για τα αρχεία που βρίσκονται μέσα σε έναν κατάλογο που έχει εισαχθεί στην βάση, στην οποία περίπτωση θα τυπώνεται το πλήρες όνομα/μονοπάτι. Για να υλοποιήσετε αυτή την λειτουργικότητα θα χρειαστείτε να χρησιμοποιήσετε τις λειτουργίες `opendir`, `readdir` και `mkdir`. Διαβάστε προσεκτικά τα αντίστοιχα `man pages` στο σύστημά σας.
- Για ακόμη περισσότερες ιδέες, επικοινωνήστε με τους διδάσκοντες.

Πακετάρισμα και αποστολή εργασίας

1. Κατασκευάστε ένα κατάλογο με όνομα `project3submit`
2. Αντιγράψτε τα `project3.c`, `objdb.h`, `objdb.c`, `Makefile` στον κατάλογο `project3submit`
3. Πακετάρτε και συμπιέστε τον κατάλογο `project3submit` κάνοντας δεξί κλικ κι επιλέγοντας `Compress here as .tar.gz`
4. Κάντε login στο `autolab`, κι επιλέξτε το `project3` του μαθήματος `ECE116-S20`
5. Αποδεχτείτε το μήνυμα ακαδημαϊκής ακεραιότητας και μετά κάντε κλικ στο `SUBMIT`.
6. Στο παράθυρο που θα εμφανιστεί εντοπίστε κι επιλέξτε το αρχείο `project3submit.tar.gz` που κατασκευάσατε ώστε να το ανεβάσετε στο `autolab`.
7. Μετά από ένα λεπτό, ανανεώσετε τη σελίδα για να δείτε το βαθμό σας στα επιμέρους τεστ.