

Πληροφορική & Τηλεπικοινωνίες

Υλοποίηση Συστημάτων Βάσεων Δεδομένων

Χειμερινό Εξάμηνο 2024-2025

Διδάσκων Θ. Μαΐλης

B+-Δέντρα

Σκοπός

Σκοπός της εργασίας αυτής είναι η κατανόηση της εσωτερικής λειτουργίας των Συστημάτων Βάσεων Δεδομένων, όσον αφορά τη διαχείριση δεδομένων σε επίπεδο μπλοκ (block). Επίσης, μέσω της εργασίας θα εξεταστεί σε ποιο βαθμό η ύπαρξη ευρετηρίων πάνω στις εγγραφές μπορεί να βελτιώσει την απόδοση ενός Συστήματος Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ). Πιο συγκεκριμένα, στα πλαίσια της 2ης εργασίας, θα υλοποιήσετε ένα σύνολο συναρτήσεων που διαχειρίζονται αρχεία δεδομένων, τα οποία έχουν δημιουργηθεί βάσει πρωτευόντων B⁺-δέντρων.

Οι συναρτήσεις που καλείστε να υλοποιήσετε αφορούν τη διαχείριση εγγραφών και ευρετηρίων. Η υλοποίησή τους θα βασιστεί υποχρεωτικά στο επίπεδο διαχείρισης μπλοκ, το οποίο παρέχεται ως βιβλιοθήκη (BF). Η διαχείριση των αρχείων ενός B⁺-δέντρου γίνεται μέσω των συναρτήσεων με το πρόθεμα BP_ τις οποίες καλείστε να υλοποιήσετε. Τα αρχεία του B⁺-δέντρου θα αποθηκεύουν εγγραφές τις οποίες πρέπει να διαχειρίζετε μέσω των κατάλληλων συναρτήσεων.

Το πρώτο μπλοκ (block) κάθε αρχείου B⁺-δέντρου θα πρέπει να περιλαμβάνει μεταδεδομένα σχετικά με το ίδιο το αρχείο. Η πληροφορία αυτή χρησιμεύει για να αναγνωριστεί αν πρόκειται πράγματι για αρχείο B⁺-δέντρου, καθώς και για να ενημερώσει τον χρήστη σχετικά με χαρακτηριστικά του B⁺-δέντρου, όπως τη ρίζα του, το ύψος του, κ.λπ. Εγγραφές για τη δοκιμή του προγράμματος παράγονται τυχαία από τη συνάρτηση randomRecord() η οποία παρέχεται μαζί με μια ενδεικτική συνάρτηση main στο eclass του μαθήματος

Συναρτήσεις BF (Block File)

Η ολοκλήρωση της εργασίας θα βασιστεί στη βιβλιοθήκη που διαχειρίζεται τα block, γνωστή ως επίπεδο block (BF), η οποία έχει παρουσιαστεί στην 1η άσκηση. Αυτή η βιβλιοθήκη λειτουργεί ως ενδιάμεση μνήμη (buffer) και διαχειριστής μνήμης (buffer manager), διευκολύνοντας την πρόσβαση στα block δίσκου και βελτιστοποιώντας τα διαβάσματα και γραψίματα στον σκληρό δίσκο.

Συναρτήσεις BP (B+-Tree)

Στην συνέχεια περιγράφονται οι συναρτήσεις που καλείστε να υλοποιήσετε και αφορούν το B⁺-δέντρο:

```
/** BP_CreateFile. Αυτή η συνάρτηση χρησιμοποιείται για τη δημιουργία και
αρχικοποίηση ενός κενού αρχείου B+ δέντρου με το όνομα fileName. Κατά τη
δημιουργία του αρχείου, προστίθενται επίσης τα μεταδεδομένα BP_info στο πρώτο
μπλοκ του αρχείου. Εάν η δημιουργία είναι επιτυχής, επιστρέφεται 0, αλλιώς
επιστρέφεται -1.**/
```

```
int BP_CreateFile(char *fileName);
```

```
/** BP_OpenFile. Αυτή η συνάρτηση ανοίγει το αρχείο B+ δέντρου με το όνομα
fileName και ανακτά τα μεταδεδομένα του. Η BP_info είναι η δομή που επιστρέφεται
```

και περιέχει τα μεταδεδομένα του B+ δέντρου. Η παράμετρος `*file_desc` είναι ο αναγνωριστικός αριθμός αρχείου που λαμβάνεται από την συνάρτηση `BF_OpenFile`.*/

```
BPLUS_INFO* BP_OpenFile(char *fileName, int *file_desc);
```

/* BP_CloseFile. Αυτή η συνάρτηση κλείνει το αρχείο B+ δέντρου που αναγνωρίζεται από το `file_desc`. Εάν είναι επιτυχής, επιστρέφει 0, διαφορετικά επιστρέφει -1. Σε περίπτωση επιτυχούς κλεισίματος, απελευθερώνει επίσης τη μνήμη που καταλαμβάνει η δομή `BP_info`, η οποία αποθηκεύει μεταδεδομένα για το B+ δέντρο.*/

```
int BP_CloseFile(int file_desc, BPLUS_INFO* info);
```

/* BP_InsertEntry: Αυτή η συνάρτηση εισάγει μία νέα εγγραφή στο B+ δέντρο. Το αρχείο αναγνωρίζεται από το `file_desc`, τα μεταδεδομένα του B+ δέντρου βρίσκονται στο `BP_info`, και η εγγραφή που θα εισαχθεί ορίζεται από τη δομή `Record`. Η συνάρτηση εντοπίζει τον κατάλληλο κόμβο δεδομένων και την εισάγει, κάνοντας τις απαραίτητες αλλαγές. Εάν η εισαγωγή είναι επιτυχής, επιστρέφει το `blockId` όπου εισήχθη η εγγραφή, διαφορετικά επιστρέφει -1. Θεωρείστε ότι το πεδίο `id` του `Record` είναι τόσο πρωτεύον κλειδί της σχέσης που αποθηκεύεται στο B+-δέντρο όσο και το κλειδί του B+-δέντρου. Συνεπώς, κάθε εισαγωγή εγγραφής με ένα `id` το οποίο ήδη υπάρχει θα πρέπει να απορρίπτεται μαζί με το σχετικό μήνυμα. */

```
int BP_InsertEntry(int file_desc, BPLUS_INFO* bplus_info, Record record);
```

/* BP_GetEntry: Αυτή η συνάρτηση αναζητά μια εγγραφή σε ένα B+ δέντρο με κλειδί που ταιριάζει με το συγκεκριμένο `id`. Ξεκινώντας από τη ρίζα, διασχίζει το B+ δέντρο για να εντοπίσει το σχετικό block δεδομένων όπου θα ήταν αποθηκευμένο το κλειδί. Αν βρεθεί μια ταιριαστή εγγραφή, η συνάρτηση θέτει το `result` να δείχνει στην εγγραφή αυτή και επιστρέφει `0` για να υποδείξει επιτυχία. Αν δεν υπάρχει τέτοια εγγραφή, η συνάρτηση θέτει το `result` σε `NULL` και επιστρέφει `-1` για να υποδείξει αποτυχία.*/

```
int BP_GetEntry(int file_desc, BPLUS_INFO* header_info, int id, Record** result);
```

Επιπλέον της υλοποίησης των συναρτήσεων, θα πρέπει να κρατάτε στην μνήμη έναν πίνακα για όλα τα ανοιχτά αρχεία. Ο ακέραιος που επιστρέφεται στην μεταβλητή `file_desc` κατά το άνοιγμα ενός αρχείου, είναι η θέση του πίνακα που αντιστοιχεί στο αρχείο που μόλις ανοίχτηκε. Σ' αυτόν τον πίνακα θα κρατάτε οτιδήποτε σχετικό κρίνετε ότι πρέπει να είναι άμεσα διαθέσιμο για κάθε ανοιχτό αρχείο (π.χ., γενικές πληροφορίες για το αρχείο, το αναγνωριστικό του αρχείου έτσι όπως έχει ανοιχτεί από το λειτουργικό σύστημα, κτλ. - χωρίς τα παραπάνω να θεωρούνται απαραίτητα). Μπορείτε να υποθέσετε ότι οποιαδήποτε στιγμή δεν θα υπάρχουν περισσότερα από `MAXOPENFILES = 20` ανοιχτά αρχεία.

Έλεγχος ορθότητας του προγράμματός σας

Η `main` που σας δίνεται σε αυτή την εργασία είναι ενδεικτική. Καλείστε να δημιουργήσετε δικές σας συναρτήσεις `main` που να αποδεικνύουν την ορθή λειτουργία του προγράμματός σας.

Αρχεία που σας δίνονται

Στα αρχεία που σας δίνονται θα βρείτε φακέλους που περιέχουν το `project` που θα πρέπει να παραδώσετε. Το `project` έχει την πιο κάτω δομή:

- **bin**: Ο κώδικας των εκτελέσιμων που δημιουργούνται

- **build:** Περιέχει όλα τα object files που δημιουργούνται κατά την μεταγλώττιση.
- **include:** Περιέχει όλα τα αρχεία κεφαλίδας που θα έχει το project σας. Θα βρείτε τα αρχεία bf.h, bp_file.h, bp_indexnode.h, bp_datanode.h.
- **lib:** Περιέχει όλες τις βιβλιοθήκες που θα χρειαστείτε για το project σας. Θα βρείτε το αρχείο libbf.so που είναι η βιβλιοθήκη για να καλείτε το επίπεδο BF.
- **src:** Τα αρχεία κώδικα (.c) τις εφαρμογής σας.
- **examples:** Σε αυτό τον φάκελο θα βρείτε ένα αρχείο main (bp_main.c) το οποίο περιέχει κώδικα που τρέχει τις συναρτήσεις που πρέπει να υλοποιήσετε.

Επίσης σας δίνετε και ένα αρχείο Makefile για να κάνετε αυτόματα compile των κωδικά σας.

Πράγματα που πρέπει να προσέξετε

- Θα πρέπει να καλείτε την συνάρτηση *BF_UnpinBlock* για κάθε BF_Block που δεν χρειάζεστε πλέον. Αν δεν τα κάνετε Unpin τότε ο buffer του επιπέδου BF γεμίζει και δεν μπορεί να “θυσιάσει” κάποιο Block γιατί όλα θα είναι ενεργά.
- Πάντα να καλείτε την συνάρτηση *BF_Block_SetDirty* για τα Block που αλλάζετε τα δεδομένα τους. Αν δεν τα κάνετε dirt τότε δεν γράφονται στον δίσκο.
- Οι συναρτήσεις που περιέχονται στη βιβλιοθήκη bp_file.h δεν θα πρέπει να αλλαχθούν. Τυχόν αλλαγές θα επιφέρουν μείωση του τελικού βαθμού.

Παράδοση εργασίας

Η εργασία αυτή είναι αυστηρά ομαδική (2 ή 3 ατόμων).

Προθεσμία παράδοσης: 10/12/2024

Γλώσσα υλοποίησης: C.

Περιβάλλον υλοποίησης: Linux (gcc 5.4+).

Περιβάλλον εξέτασης: Η εργασία θα εξεταστεί στα linux του di.uoa και καλείστε να έχετε ελέγξει ότι εκτελείται ορθώς στο συγκεκριμένο περιβάλλον.

Παραδοτέα

- Όλος ο φάκελος bPlus μαζί με αρχείο Makefile που θα κάνει link τον κώδικά σας με την ήδη υπάρχουσα main αλλά και με τις υπόλοιπες που θα υλοποιήσετε. Ο φάκελος αυτός να περιέχεται σε zip αρχείο αφού πρώτα έχουν αφαιρεθεί τυχόν **αρχεία δεδομένων** **`.db`** που δημιουργήθηκαν.
- Ένα README που θα εξηγεί τις σχεδιαστικές επιλογές ή τυχόν παραδοχές που έχετε κάνει. Επίσης θα αναφέρεται τυχόν δυσλειτουργίες του προγράμματος σας που έχετε παρατηρήσει. Η παράδοση θα γίνει ανά ομάδα εργασίας.