

Υλοποίηση Συστημάτων Βάσεων Δεδομένων

Εργασίες 1 & 2

Κωνσταντίνος Χούσος

AM: 1115202000215

Αναστάσιος-Φαίδων Σεϊτανίδης

AM: 1115202000179

ΕΡΓΑΣΙΑ 1

Αρχείο Σωρού

Συναρτήσεις

Οι παρακάτω συναρτήσεις εκμεταλλεύονται τις συναρτήσεις επιπέδου block **BF_*** για την υλοποίησή τους, προσθέτοντας ωστόσο παραπάνω λειτουργικότητα.

- **HP_CreateFile:** Δημιουργεί το αρχείο με όνομα **filename**, δημιουργεί το block 0 και αρχικοποιεί τις τιμές του όσον αφορά τα **HP_info** και **HP_block_info**. Κάνει το block 0 dirty και το κάνει unpin. Τέλος, κλείνει το αρχείο.
- **HP_OpenFile:** Ανοίγει το αρχείο σωρού με όνομα **filename**. Ελέγχει αν το αρχείο πρόκειται για αρχείο κατακερματισμού ή όχι. Αν όχι, συνεχίζει. Καθώς κάθε φορά που καλείται η **BF_OpenFile** το **fileDesc** αλλάζει, ενημερώνουμε κάθε φορά το **HP_info** του block 0. Κάνουμε dirty κι unpin το block 0, κρατώντας ωστόσο το **HP_info** στον σωρό κι επιστρέφοντάς το.
- **HP_CloseFile:** Βρίσκει πόσα blocks έχει το αρχείο σωρού. Σε ένα for loop κάνει το καθένα unpin. Κλείνει το αρχείο κι αποδεσμεύει την μνήμη του **HP_info** που υπάρχει στον σωρό.
- **HP_InsertEntry:** Βρίσκει το τελευταίο block του αρχείου σωρού. Ελέγχει αν υπάρχει αρκετός χώρος σε αυτό για την εγγραφή μιας πλειάδας. Αν ναι, αντιγράφει την πλειάδα σε αυτό. Αν όχι, δημιουργεί ένα νέο block στο τέλος του αρχείου σωρού, γράφει σε αυτό την πλειάδα και αρχικοποιεί τα metadata του. Έπειτα ενημερώνει το **HP_info** για το νέο block και το κάνει unpin. Τέλος, επιστρέφει τον αναγνωριστικό αριθμό του block στο οποίο έγινε τελικά η εγγραφή.
- **HP_GetAllEntries:** Η συνάρτηση επιστρέφει μία μεταβλητή **int read_blocks**. Αυτή αρχικοποιείται με -1 και ανανεώνεται όταν βρούμε ένα record με **id = value**. Αν επιστραφεί και δεν έχει αλλάξει, σημαίνει πως υπήρξε κάποιο σφάλμα. Διατρέχει όλα τα blocks του αρχείου σωρού, βρίσκει για το καθένα πόσες εγγραφές υπάρχουν και τυπώνει αυτές που έχουν **id = value**.

Δομή Αρχείου Σωρού

- Δομή **HP_info**

Ένα αρχείο σωρού χαρακτηρίζεται από τα εξής στοιχεία:

- **int fileDesc:** Ο αναγνωριστικός αριθμός ανοίγματος αρχείου από το επίπεδο block.

- **int lastBlockDesc:** Ο αναγνωριστικός αριθμός του τελευταίου σε αριθμό block του αρχείου σωρού.
 - **int isHT:** Ένα flag που δηλώνει αν το αρχείο είναι αρχείο κατακερματισμού ή όχι. Στην περίπτωση του αρχείου σωρού αρχικοποιείται με 0.
- Δομή **HP_block_info**
Ένα block ενός αρχείου σωρού χαρακτηρίζεται από τα εξής στοιχεία:
 - **int blockDesc:** Ο αναγνωριστικός αριθμός του block στην αρίθμηση των block του αρχείου σωρού.
 - **int recsNum:** Το πλήθος εγγεγραμμένων πλειάδων στο συγκεκριμένο block.
 - **int nextBlock:** Ο αναγνωριστικός αριθμός του επόμενου block στην αρίθμηση των block του αρχείου σωρού. Αν δεν υπάρχει τότε έχει την τιμή -1, με την οποία αρχικοποιείται.

Σημειώσεις

Το **HP_ERROR** ορίζεται σε -1, καθώς δεν έχει αρχικοποιηθεί κι αλλιώς δεν δουλεύει η συνάρτηση **CALL_BF**.

Αρχείο Κατακερματισμού

Συναρτήσεις

Οι παρακάτω συναρτήσεις εκμεταλλεύονται τις συναρτήσεις επιπέδου block **BF_*** για την υλοποίησή τους, προσθέτοντας ωστόσο παραπάνω λειτουργικότητα.

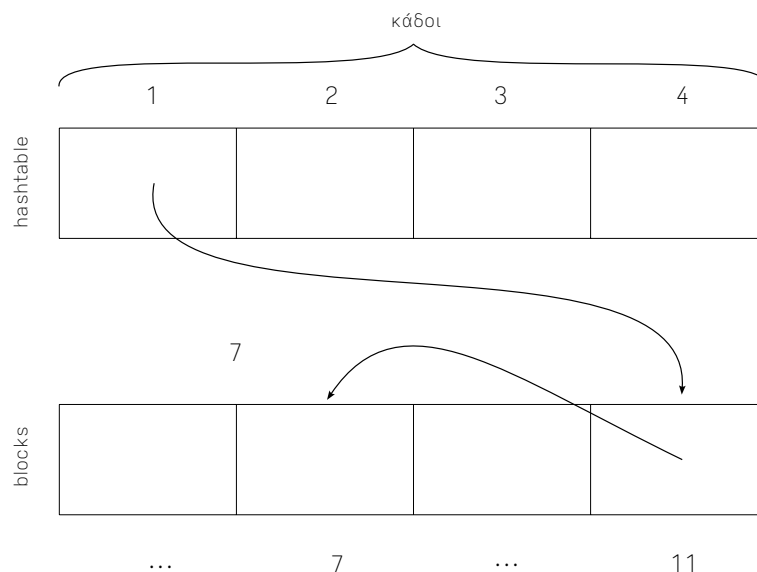
- **HT_CreateFile:** Δημιουργεί το αρχείο με όνομα **filename**, δημιουργεί το block 0 κι αρχικοποιεί τις τιμές του όσον αφορά τα **HT_info** και **HT_block_info**. Κάνει το block 0 dirty και το κάνει unpin. Τέλος, κλείνει το αρχείο.
- **HT_OpenFile:** Ανοίγει το αρχείο σωρού με όνομα **filename**. Ελέγχει αν το αρχείο πρόκειται για αρχείο κατακερματισμού ή όχι. Αν ναι, συνεχίζει. Αρχικοποιεί το hashtable που θα υπάρχει στην μνήμη. Καθώς κάθε φορά που καλείται η **BF_OpenFile** το **fileDesc** αλλάζει, ενημερώνουμε κάθε φορά το **HT_info** του block 0. Κάνουμε dirty κι unpin το block 0, κρατώντας ωστόσο το **HT_info** στον σωρό κι επιστρέφοντάς το.
- **HT_CloseFile:** Βρίσκει πόσα blocks έχει το αρχείο σωρού. Σε ένα for loop κάνει το καθένα unpin. Κλείνει το αρχείο κι αποδεσμεύει την μνήμη του **HT_info** που υπάρχει στον σωρό, καθώς και την μνήμη του hashtable.
- **HT_InsertEntry:** Βρίσκει τον κάδο στον οποίο πρέπει να καταγραφεί η εγγραφή. Αν ο κάδος δεν έχει κάποιο block, τότε δημιουργεί ένα και το συσχετίζει με τον κάδο μέσω του hashtable. Βρίσκουμε αν στο πιο πρόσφατο block του κάδου χωράει η εγγραφή. Αν ναι, την γράφει εκεί. Αν όχι, δημιουργεί ένα νέο block για τον κάδο—το οποίο γίνεται το πιο πρόσφατό του—και γράφει εκεί την εγγραφή.
- **HT_GetAllEntries:** Η συνάρτηση επιστρέφει μία μεταβλητή **int read_blocks**. Αυτή αρχικοποιείται με -1 και ανανεώνεται όταν βρούμε ένα record με **id = value**. Αν επιστραφεί και δεν έχει αλλάξει, σημαίνει πως υπήρξε κάποιο σφάλμα. Βρίσκει τον κάδο που περιέχει εγγραφές με **id = value** και διατρέχει όλα τα blocks του, από το πιο πρόσφατο προς το πρώτο.

Δομή Αρχείου Κατακερματισμού

- Δομή **HT_info**

Ένα αρχείο σωρού χαρακτηρίζεται από τα εξής στοιχεία:

- **int fileDesc**: 0 αναγνωριστικός αριθμός ανοίγματος αρχείου από το επίπεδο block.
- **int lastBlockDesc**: 0 αναγνωριστικός αριθμός του τελευταίου σε αριθμό block του αρχείου σωρού.
- **int isHT**: Ένα flag που δηλώνει αν το αρχείο είναι αρχείο κατακερματισμού ή όχι. Στην περίπτωση του αρχείου κατακερματισμού αρχικοποιείται με 1.
- **long int numBuckets**: Το πλήθος των κάδων που θα έχει το αρχείο κατακερματισμού.
- **int *hashtable**: Ένας πίνακας μεταβλητού μεγέθους σε int. Το hashtable του αρχείου σωρού αναπαριστάται ως εξής: Η κάθε θέση του πίνακα hashtable αναπαριστά έναν κάδο. Δηλαδή για παράδειγμα, το **hashtable[2]** δηλώνει τον κάδο 2. Το περιεχόμενο της κάθε θέσης όμως είναι ένας int, ο οποίος ταυτίζεται με το **blockDesc** του block του κάδου που προστέθηκε πιο πρόσφατα σε αυτόν. Ένα παράδειγμα της μοντελοποίησης αυτής υπάρχει στο σχήμα 1.



Σχήμα 1: Παράδειγμα μοντελοποίησης hashtable. Αν ο κάδος 1 αποτελείται από το block 7, τότε **hashtable[1] == 7**. Αν το block 7 κάποια στιγμή γεμίσει, τότε για να προστεθεί μια νέα εγγραφή στον κάδο 1 θα δημιουργηθεί ένα νέο block στο τέλος του αρχείου. Έστω ότι το νέο block έχει αριθμό—κι άρα **blockDesc**—11. Πλέον το **hashtable[1]** θα ενημερωθεί και θα αποκτήσει την τιμή 11, τον αριθμό δηλαδή του πιο πρόσφατου block του. Στο **HT_block_info** του block 11 ο αναγνωριστικός αριθμός **prevBlockDesc** θα αλλάξει από -1 και θα γίνει 7, έτσι ώστε να δείχνει στο προηγούμενο στη σειρά block. Άρα για κάθε κάδο έχουμε πρόσβαση σε όλα τα blocks του σειριακά.

- Δομή **HT_block_info**

Ένα block ενός αρχείου σωρού χαρακτηρίζεται από τα εξής στοιχεία:

- **int blockDesc**: 0 αναγνωριστικός αριθμός του block στην αρίθμηση των block του αρχείου σωρού.

- **int prevBlockDesc**: Ο αναγνωριστικός αριθμός του προηγούμενου block του κάδου. Αρχικοποιείται με -1 εφόσον δεν υπάρχει προηγούμενο block. Στο παράδειγμα του σχήματος 1 το **prevBlockDesc** του block 11 θα είναι ο αριθμός 7, ενώ το **prevBlockDesc** του block 7 θα είναι -1.
- **int recsNum**: Το πλήθος εγγεγραμμένων πλειάδων στο συγκεκριμένο block.

ΣΗΜΕΙΩΣΕΙΣ**COMMENT**

TODO Στην `SHT_SecondaryGetAllEntries` θα έχουμε μια δική της μεταβλητή που θα μετράει τα `read blocks` και θα προσθέτουμε και αυτό που επιστρέφει η `HT_GetAllEntries`.