

# Υλοποίηση Συστημάτων Βάσεων Δεδομένων Εργασία 2 Τεκμηρίωση Κώδικα

Λάμπρου Ιωάννης -1115201400088  
Χατζηδάκης Παφάηλ -1115201400248

3 Δεκεμβρίου 2017

Για την εργασία αυτή, υλοποιήθηκαν πλήρως όλες οι συναρτήσεις επιπέδου AM που ζητήθηκαν στην εκφώνηση και συμπεριφέρονται σύμφωνα με αυτή.

Κάθε αρχείο ευρετηρίου B+ δέντρου αποτελείται από 3 είδη μπλοκ. Αρχικά, η ρίζα και τα κλαδιά του δέντρου είναι μπλοκ ευρετηρίου, (index block) έχοντας αρχικά ".ib", 4 bytes, έπειτα τον αριθμό των κλειδιών που είναι αποθηκευμένα σε αυτό και τέλος, δείκτη σε μπλοκ (αριθμός του μπλοκ) και κλειδί τα οποία εναλλάσσονται ώσπου και τελειώνει σε δείκτη σε μπλοκ (ο αριθμός των δεικτών είναι κατά ένα μεγαλύτερος από τον αριθμό των κλειδιών).

Τα φύλλα του δέντρου είναι μπλοκ δεδομένων (περιέχουν εγγραφές) (data block) αρχίζοντας από ".db", 4 bytes. Μέσα έχουν αποθηκευμένο τον αριθμό των εγγραφών που υπάρχουν μέσα τους, έπειτα έναν δείκτη (αριθμό μπλοκ) στο επόμενο (πιο δεξιό) μπλοκ δεδομένων, ακολουθούμενο από τις εγγραφές που έχουμε εισάγει, (attribute1 και attribute2)

Τέλος, το πρώτο μπλοκ κάθε αρχείου είναι μπλοκ μεταδεδομένων και χρησιμοποιείται για να αποθηκεύει τον τύπο του αρχείου (index file (".if", 4 bytes), τα attribute types, lengths για κάθε έναν από τους τύπους δεδομένων που θα αποθηκεύονται στο αρχείο, καθώς και τον αριθμό του μπλοκ της ρίζας του B+ δέντρου και τον αριθμό του πρώτου μπλοκ δεδομένων (αριστερότερο).

Για την υλοποίηση της άσκησης επίσης χρησιμοποιήθηκαν δύο πίνακες από extern structs, οι Filemeta OpenIndexes, SearchData OpenSearches, οι οποίες και χρησιμοποιούνται για την αποθήκευση χρήσιμων δεδομένων για τις Insert και Scan αντίστοιχα. Στην δομή OpenIndexes αποθηκεύεται ο file descriptor που δόθηκε από το σύστημα για την πρόσβαση σε αυτό, το όνομα του αρχείου, τα attribute types και lengths για κάθε έναν από τους τύπους δεδομένων που θα αποθηκεύονται στο αρχείο, καθώς και ο αριθμός του μπλοκ της ρίζας του B+ δέντρου και ο αριθμός του πρώτου μπλοκ δεδομένων (αριστερότερο). Στην δομή OpenSearches αποθηκεύεται η έξοδος της AM\_Find\_Next\_Entry, η θέση στον πίνακα OpenIndexes του αρχείου στο οποίο γίνεται η αναζήτηση, η πράξη που γίνεται, το μπλοκ και η θέση στην οποία βρίσκεται η AM\_Find\_Next\_Entry κατά τη διαδικασία της εκτύπωσης, και ένα κλειδί, ανάλογα με την πράξη.

Για την υλοποίηση της Insert σημειώνεται χρησιμοποιήθηκε μια αναδρομική συνάρτηση, η rec\_trav\_insert, η οποία, αναδρομικά κατεβαίνει το B+ δέντρο και όταν

φτάσει σε φύλλο, (data block) τότε και κάνει την εισαγωγή της δεδομένης εγγραφής, και, αν χρειαστεί να δημιουργηθεί κάποιο καινούργιο μπλοκ, τότε και περνάει τις απαραίτητες πληροφορίες σε ανώτερο επίπεδο του δέντρου με τιμή επιστροφής (στον εαυτό της από τον οποίο και κλήθηκε).

Ακόμα, για την ομαλή λειτουργία του `AM_Find_Next_Entry`, όταν δεν βρίσκεται κάποιο κλειδί, τότε και η τιμή του `OpenSearches[].curr_pos` γίνεται ίση με -2, έτσι ώστε όταν αυτό εντοπίζεται από την `AM_Find_Next_Entry`, τότε και αυτή έχει ως έξοδο `NULL` και αλλάζει τον κωδικό λάθους σε `AME_EOF`

Όλες οι λειτουργίες του ευρετηρίου (εισαγωγές ή εξαγωγές εγγραφών) δουλεύουν μια χαρά ((`am_main1`, `am_main3`) αλλά φαίνεται, στην `am_main2` να μην μπορεί να διαβάσει από το μπλοκ των μεταδεδομένων, τον αριθμό του μπλοκ της ρίζας του δέντρου. Αυτό το πρόβλημα φαίνεται να λύνεται όμως, για κάποιο λόγο, αν αντιγράψουμε τον κώδικα της `am_main2` στην `am_main1`.

Η ανάπτυξη και εκτέλεση-δοκιμή της εργασίας έγινε σε περιβάλλον Ubuntu 16.04.