

Ονοματεπώνυμο: Γιάννης Καλογερόπουλος A.M.:
1115201500057

Ονοματεπώνυμο: Ηρακλή Κότσι A.M.:
1115201500073

Ονοματεπώνυμο: Ηλίας Παπασωτηρίου A.M.:
1115201500123

Δομή αρχείων:

Στον φάκελο src/ υπάρχουν το AM.c και το functions.c

AM.c: περιέχει τις βασικές συναρτήσεις που αναφέρονται στην εκφώνηση και την print_data()(ως βοηθητική).

functions.c: περιέχει τις πρόσθετες συναρτήσεις που χρειάζονται για την υλοποίηση του προγράμματος

Στον φάκελο include/ προστέθηκε το functions.h με τις δηλώσεις του functions.c και στο AM.h προστέθηκαν κάποιοι κωδικοί λάθους

Το Makefile είναι ανανεωμένο.

Πληροφορίες κώδικα:

Υπάρχουν 2 struct το DataIndex και το IndexScan. Το DataIndex περιέχει πληροφορίες αρχείου (όπως filedesc, attrType1, attrLength1, attrType2, attrLength2, το όνομα του αρχείου και την θέση της ρίζας). Το IndexScan περιέχει πληροφορίες για τα αρχεία σαρώσεις, συγκεκριμένα έχει: filedesc, operator(=,!=,> κτλ.) , blocknum και indexpos(για την θέση του πεδίου) και το fD που είναι η θέση του ανοιχτού αρχείου στον πίνακα DataIndex.

Επίσης έχουμε 2 global πίνακες ,όπως ζητείτε, για τα ανοιχτά αρχεία (OpenIndex) και για τις ανοιχτές σαρώσεις (OpenIndexScan).

Ένα μπλοκ ευρετηρίου έχει την μορφή:

'I' | δείκτης πατέρα | αριθμός κλειδιών |
...δεδομένα ...

Ένα μπλοκ δεδομένων έχει την μορφή:

'D' | δείκτης πατέρα | αριθμός κλειδιών | ...δεδομένα
...

Στο AM.c σειρά 278 το 2ο όρισμα της συνάρτησης είναι 1 ενώ θα έπρεπε να είναι ο αναγνωριστικός αριθμός μπλόκ της ρίζας. Το θέσαμε έτσι γιατί έτσι εμφανίζονταν τα αποτελέσματα.

- Η συνάρτηση `AM_Init` αρχικοποιεί τους 2 προαναφερθέντος πίνακες.
- Η `AM_CreateIndex` φτιάχνει το πρώτο μπλοκ με τα δεδομένα που χρειάζεται ένα B+ δέντρο (χαρακτηριστικό ότι είναι B+, `attrType1`, `attrLength1`, `attrType2`, `attrLength2` και την θέση της ρίζας που επειδή είναι άδειο την αρχικοποιούμε με -1).
- Η `AM_DestroyIndex` ελέγχει αν υπάρχει το αρχείο με όνομα `fileName`. Αν ναι, τότε με την συνάρτηση `remove(fileName)` διαγράφει τον φυσικό χώρο του αρχείου από το δίσκο, αλλιώς επιστρέφει κωδικό λάθους.
- Η `AM_OpenIndex` ανοίγει το αρχείο που ζητείται ελέγχει αν είναι αρχείο B+ δέντρου και ψάχνει να του βρεί μια θέση στον πίνακα με τα ανοιχτά αρχεία (`OpenIndex`). Αν ο πίνακας έχει χώρο τότε του περνάει ένα `DataIndex` με όλες τις πληροφορίες (οι οποίες βρίσκονται στο πρώτο μπλοκ με τα δεδομένα).
- Η `AM_CloseIndex` βρίσκει την θέση του αρχείου στον πίνακα και ενημερώνει το πρώτο μπλοκ του αρχείου με τις αλλαγές που έχουν γίνει (δηλαδή αν έχει γίνει αλλαγή της ρίζας).
- Η `AM_InsertEntry` υπολογίζει στην αρχή τον συνολικό αριθμό με τα κλειδιά που χωράει ένα μπλοκ δεδομένων (`totalData`) και ευρετηρίου (`totalIndex`). Έπειτα αν το δέντρο δεν έχει κανένα στοιχείο (άρα είναι εντελώς

άδειο) τότε καλείται η συνάρτηση `Root_Creation` που δημιουργεί και αρχικοποιεί 2 μπλοκ ένα για την ρίζα (1το ευρετήριο) και το πρώτο μπλοκ φύλου που θα έχει την πρώτη εισαγωγή. **Σημαντικό** , ο πρώτος "δείκτης" του ευρετηρίου περιέχει την θέση του μπλοκ της πρώτης εισαγωγής. Όταν υπάρχει μόνο ένα μπλοκ με δεδομένα δεν υπάρχει κανένα κλειδί στο ευρετήριο και ο δείκτης που δείχνει στα δεδομένα είναι ο αριστερά αριστερά. Αν το δέντρο δεν είναι άδειο καλείται η `search_position` που βρίσκει τον αριθμό του μπλοκ που πρέπει να τοποθετηθεί το νέο κλειδί-δεδομένο. Μετά καλείται η `Insert` που βλέπει εαν έχει χώρο το μπλοκ που πρέπει να μπει το νέο δεδομένο και αν έχει καλεί την `InsertWithSorting` που το εισάγει ταξινομημένα στο μπλοκ αλλιώς καλείται η `break_it` που σπάει το μπλοκ φύλου και εισάγει (`InsertForIndexWithSorting`) το νέο κλειδί στο πατέρα-ευρετήριο αν έχει χώρο αλλιώς καλείται η `break_Index` για να σπάσει το ευρετήριο που είναι αναδρομική και ελέγχει και την περίπτωση να είναι ρίζα. Τέλος έχουμε σε σχόλια την `print_data` που εκτυπώνει τα δεδομένα που έχουν εισαχθεί.

- Η `AM_OpenIndexScan` βρίσκει το μπλοκ ρίζας του δέντρου και έπειτα φτάνει στο μπλόκ δεδομένων με τις μικρότερες τιμές. Χρησιμοποιεί την `switch` για τις διάφορες τιμές του `op` . Μετά ψάχνει τα πεδία-κλειδιά ανάλογα με την συνθήκη και σταματάει στις ανάλογες περιπτώσεις. Επίσης μετακινείται από μπλοκ σε μπλοκ με τον αριθμό του επόμενου μπλοκ με τα δεδομένα του βρίσκεται στο τέλος κάθε μπλοκ. Τέλος αποθηκεύει στον πίνακα `OpenIndexScan` την θέση που βρήκε την πρώτη κατάλληλη τιμή και επιστρέφει την θέση του πίνακα. (Να σημειωθεί ότι ως ρίζα δίνεται το μπλόκ με `αρ.μπλοκ=1` λόγω του ότι καποιες λειτουργίες δεν εκτελούνται σωστά και δεν εμφανίζεται κανένα αποτέλεσμα στην συνέχεια.)
- Η `AM_FindNextEntry` μέσω της θέσης του πίνακα βρίσκει την θέση του στοιχείου που βρεθηκε απ'την

AM_OpenIndexScan και συνεχίζει από εκεί και μετά το ψάξιμο των επόμενων τιμών.

- Η AM_CloseIndexScan αποδεσμεύει την θέση του πίνακα με τα αρχεία σάρωσης αν υπάρχει ανοιχτό το αρχείο αλλιώς εμφανίζει κατάλληλο μήνυμα.
- Η AM_Close() κλείνει πρώτα όλες τις ανοικτές σαρώσεις και έπειτα τα ανοικτά αρχεία για να καταστραφούν σωστά οι δομές που είχαν αρχικοποιηθεί.

Περισσότερα σχόλια βρίσκονται στα αρχεία.