Πληροφορική & Τηλεπικοινωνίες

Κ18 - Υλοποίηση Συστημάτων Βάσεων Δεδομένων
Εαρινό Εξάμηνο 2010 – 2011

Δ. Γουνόπουλος – Ι. Ιωαννίδης

Άσκηση 1: Συγχώνευση Εγγραφών Δυαδικών Αρχείων
Προθεσμία: 2 Μαΐου 2011, 5:00μμ

Σκοπός της εργασίας είναι η κατανόηση της εσωτερικής λειτουργίας των Συστημάτων Βάσεων Δεδομένων ως προς τη διαχείριση σε επίπεδο μπλοκ (block) αλλά και σε επίπεδο εγγραφών. Επίσης, αφορά μια βασική λειτουργία που πραγματοποιείται συχνά στις βάσεις δεδομένων, που είναι η *συγχώνευση αρχείων* (merge). Τέλος, η εργασία θα λειτουργήσει και ως ένα "ζέσταμα" στη γλώσσα προγραμματισμού *C / C++*.

Στα πλαίσια της εργασίας αυτής καλείστε να υλοποιήσετε συναρτήσεις με συγκεκριμένη λειτουργικότητα, ώστε να συγχωνεύονται δυαδικά αρχεία εγγραφών. Πιο συγκεκριμένα, θα υλοποιήσετε ένα πλήθος συναρτήσεων ώστε, δοθέντων των ονομάτων δύο ή περισσότερων αρχείων (*αρχεία προέλευσης*), να δημιουργείται ένα νέο αρχείο (*αρχείο προορισμού*), όπου θα τοποθετούνται ταξινομημένες οι εγγραφές. Η ταξινόμηση θα γίνεται ως προς συγκεκριμένο πεδίο, που θα δίνεται ως όρισμα κατά την κλήση της συνάρτησης. Τα αρχεία προέλευσης θα είναι ήδη ταξινομημένα το καθένα ως προς το αντίστοιχο πεδίο. Η ζητούμενη λειτουργικότητα πρέπει να υλοποιηθεί πάνω από το επίπεδο διαχείρισης μπλοκ (block), το οποίο σας δίνεται έτοιμο.

Τόσο τα αρχεία που δίνονται, όσο και αυτό που θα δημιουργήσετε μέσω της συγχώνευσης, θα έχουν *εγγραφές,* οι οποίες εννοιολογικά αντιστοιχούν σε

```
typedef struct{
        int id;
        char name[MAXNAME];
        char surname[MAXNAME];
        float avgPoints;
}Record;
```

Οι εγγραφές μέσα στα αρχεία είναι αποθηκευμένα ως ακολουθίες από byte, η μία μετά την άλλη. Τα πεδία κάθε εγγραφής έχουν τη σειρά που δίνεται προηγουμένως και τους αντίστοιχους τύπους δεδομένων. Κατά συνέπεια, οφείλετε να διαχειρίζεστε τα πεδία με τις κατάλληλες συναρτήσεις που σας δίνονται από τη C / C++.

Ακολουθούν κάποιες ενδεικτικές εγγραφές, που μπορεί να υπάρχουν στα αρχεία εισόδου. Τα εισαγωγικά (") **δεν** υπάρχουν στην πραγματική εγγραφή. Εμφανίζονται εδώ, για να δηλώσουμε ότι πρόκειται για συμβολοσειρά.

{15, "Giorgos", "Papadopoulos", 7.34 }
{4, "Tasos", "Politis", 8.12}
{300, "Yannis", "Stratakis", 6.89 }

Συναρτήσεις BF (Block File)

Στη συνέχεια, περιγράφονται οι συναρτήσεις που αφορούν το επίπεδο από block, πάνω στο οποίο θα βασιστείτε για την υλοποίηση των συναρτήσεων που ζητούνται. Η υλοποίηση των συναρτήσεων αυτών θα δοθεί έτοιμη με τη μορφή βιβλιοθήκης. Στο αρχείο κεφαλίδας **BF.h** που θα σας δοθεί υπάρχουν τα πρωτότυπα των συναρτήσεων μαζί με σχόλια για το πώς δουλεύουν και το μέγεθος ενός μπλοκ που είναι ***BF_BLOCK_SIZE***, ίσο με ***1024*** bytes.
Θα δοθεί επίσης και μία ενδεικτική main() που δείχνει πως μπορείτε να χρησιμοποιήσετε το επίπεδο **BF**.

void BF_Init()

Με τη συνάρτηση *BF_Init* πραγματοποιείται η αρχικοποίηση του επιπέδου BF.

int BF_CreateFile(*char* filename /* όνομα αρχείου */* ***)***

Η συνάρτηση *BF_CreateFile* δημιουργεί ένα αρχείο με όνομα *filename* το οποίο αποτελείται από block. Αν το αρχείο υπάρχει ήδη το παλιό αρχείο διαγράφεται. Σε περίπτωση επιτυχούς εκτέλεσης της συνάρτησης επιστρέφεται 0, ενώ σε περίπτωση αποτυχίας, επιστρέφεται ένας αρνητικός αριθμός. Αν θέλετε να δείτε το είδος του λάθους μπορείτε να καλέσετε τη συνάρτηση *BF_PrintError*.

int BF_OpenFile(*char* filename /* όνομα αρχείου */* ***)***

Η συνάρτηση *BF_OpenFile* ανοίγει ένα υπάρχον αρχείο από block με όνομα *filename*. Σε περίπτωση επιτυχίας, επιστρέφεται το αναγνωριστικό του αρχείου, ενώ σε περίπτωση αποτυχίας, επιστρέφεται ένας αρνητικός αριθμός. Αν θέλετε να δείτε το είδος του λάθους μπορείτε να καλέσετε τη συνάρτηση *BF_PrintError*.

int BF_CloseFile(*int fileDesc /* αναγνωριστικό αρχείου block */* ***)***

Η συνάρτηση *BF_CloseFile* κλείνει το ανοιχτό αρχείο με αναγνωριστικό αριθμό *fileDesc*. Σε περίπτωση επιτυχίας επιστρέφει 0, ενώ σε περίπτωση αποτυχίας, επιστρέφεται ένας αρνητικός αριθμός. Αν θέλετε να δείτε το είδος του λάθους μπορείτε να καλέσετε τη συνάρτηση *BF_PrintError*.

int BF_GetBlockCounter(*int fileDesc /* αναγνωριστικό αρχείου block */* ***)***

Η συνάρτηση *Get_BlockCounter* δέχεται ως όρισμα τον αναγνωριστικό αριθμό *fileDesc* ενός ανοιχτού αρχείου από block και βρίσκει τον αριθμό των διαθέσιμων block του, τον οποίο και επιστρέφει σε περίπτωση επιτυχούς εκτέλεσης. Σε περίπτωση αποτυχίας, επιστρέφεται ένας αρνητικός αριθμός. Αν θέλετε να δείτε το είδος του λάθους μπορείτε να καλέσετε τη συνάρτηση *BF_PrintError*.

int BF_AllocateBlock(*int fileDesc /* αναγνωριστικό αρχείου block */* ***)***

Με τη συνάρτηση *BF_AllocateBlock* δεσμεύεται ένα καινούριο block για το αρχείο με αναγνωριστικό αριθμό *fileDesc*. Το νέο block αρχικοποιείται με μηδενικά και δεσμεύεται πάντα στο τέλος του αρχείου, οπότε ο αριθμός του block είναι *BF_GetBlockCounter(fileDesc) - 1*. Σε περίπτωση επιτυχίας επιστρέφει 0, ενώ σε περίπτωση αποτυχίας, επιστρέφεται ένας αρνητικός αριθμός. Αν θέλετε να δείτε το είδος του λάθους μπορείτε να καλέσετε τη συνάρτηση *BF_PrintError*.

int BF_ReadBlock(
int fileDesc, / αναγνωριστικό αρχείου block */*
int blockNumber, / ο αριθμός ενός δεσμευμένου block μέσα στο αρχείο */*
*void** block /* δείκτης στα δεδομένα του block (αναφορά) */* ***)***

Η συνάρτηση *BF_ReadBlock* βρίσκει το block με αριθμό *blockNumber* του ανοιχτού αρχείου με αναγνωριστικό αριθμό *fileDesc*. Η μεταβλητή *block* αποτελεί ένα δείκτη στα δεδομένα του block, το οποίο θα πρέπει να έχει δεσμευτεί. Σε περίπτωση επιτυχίας, επιστρέφει 0, ενώ σε περίπτωση αποτυχίας, επιστρέφεται ένας αρνητικός αριθμός. Αν θέλετε να δείτε το είδος του λάθους μπορείτε να καλέσετε τη συνάρτηση *BF_PrintError*.

int BF_WriteBlock(
int fileDesc, / αναγνωριστικό αρχείου block */*
int blockNumber / ο αριθμός ενός δεσμευμένου block μέσα στο αρχείο */* ***)***

Η συνάρτηση *BF_WriteBlock* γράφει στο δίσκο το δεσμευμένο block με αριθμό *blockNumber* του ανοιχτού αρχείου με αναγνωριστικό αριθμό *fileDesc*. Σε περίπτωση επιτυχίας, επιστρέφει 0, ενώ σε περίπτωση αποτυχίας, επιστρέφεται ένας αρνητικός αριθμός. Αν θέλετε να δείτε το είδος του λάθους μπορείτε να καλέσετε τη συνάρτηση *BF_PrintError*.

void BF_PrintError(*char* message /* μήνυμα προς εκτύπωση */* ***)***

Η συνάρτηση *BF_PrintError* βοηθά στην εκτύπωση των σφαλμάτων που δύναται να υπάρξουν με την κλήση συναρτήσεων του επιπέδου αρχείου block. Εκτυπώνεται στο stderr το *message* ακολουθούμενο από μια περιγραφή του πιο πρόσφατου σφάλματος.

Συναρτήσεις Merge

Στη συνέχεια, περιγράφονται οι συναρτήσεις που καλείστε να υλοποιήσετε στα πλαίσια της εργασίας αυτής και που αφορούν το ταξινομημένο αρχείο.

int Sorted_CreateFile(*char *fileName /* όνομα αρχείου */* ***)***

Η συνάρτηση Sorted_CreateFile χρησιμοποιείται για τη δημιουργία και κατάλληλη αρχικοποίηση ενός άδειου αρχείου με όνομα *fileName*. Σε περίπτωση που εκτελεστεί επιτυχώς, επιστρέφεται 0, ενώ σε διαφορετική περίπτωση -1.

int Sorted_OpenFile(*char *fileName /* όνομα αρχείου */* ***)***

Η συνάρτηση Sorted_OpenFile ανοίγει το αρχείο με όνομα *filename* και διαβάζει από το πρώτο μπλοκ την πληροφορία που αφορά το αρχείο. Επιστρέφει τον αναγνωριστικό αριθμό ανοίγματος αρχείου, όπως αυτός επιστράφηκε από το επίπεδο διαχείρισης μπλοκ ή -1 σε περίπτωση σφάλματος.

int Sorted_CloseFile(*int fileDesc /* αναγνωριστικός αριθμός ανοίγματος αρχείου */* ***)***

Η συνάρτηση *Sorted_CloseFile* κλείνει το αρχείο που προσδιορίζεται από τον αναγνωριστικό αριθμό ανοίγματος *fileDesc*. Σε περίπτωση που εκτελεστεί επιτυχώς, επιστρέφεται 0, ενώ σε διαφορετική περίπτωση -1.

int Sorted_InsertFirstEntry (
int fileDesc, / αναγνωριστικός αριθμός ανοίγματος αρχείου */*
Record record / δομή που προσδιορίζει την εγγραφή */* ***)***

Η συνάρτηση *Sorted_InsertEntry* χρησιμοποιείται για την εισαγωγή της πρώτης εγγραφής στο ταξινομημένο αρχείο. Ο αναγνωριστικός αριθμός ανοίγματος του αρχείου δίνεται με την *fileDesc* ενώ η εγγραφή προς εισαγωγή προσδιορίζεται από τη δομή *record*. Η εγγραφή πρέπει να ειναι η πρώτη εγγραφή στο αρχείο. Σε περίπτωση που εκτελεστεί επιτυχώς, επιστρέφεται 0, ενώ σε διαφορετική περίπτωση -1.

void Sorted_GetAllEntries(
int fileDesc, / αναγνωριστικός αριθμός ανοίγματος αρχείου */*
char fieldName, /* όνομα του πεδίου για το οποίο γίνεται ο έλεγχος */*
*void *value /* τιμή του πεδίου προς σύγκριση */* ***)***

Η συνάρτηση αυτή χρησιμοποιείται για την εκτύπωση όλων των εγγραφών που υπάρχουν στο ταξινομημένο αρχείο οι οποίες έχουν τιμή στο πεδίο με όνομα *fieldName* ίση με *value*. Το *fileDesc* είναι ο αναγνωριστικός αριθμός ανοίγματος του αρχείου, όπως αυτός έχει επιστραφεί από το επίπεδο διαχείρισης μπλοκ. Η παράμετρος *fieldName* μπορεί να παίρνει μία από τις εξής τιμές: "***id***", "***name***", "***surname***" ή "***avgPoints*** ", αναφερόμενη στα αντίστοιχα πεδία μιας εγγραφής.
Για κάθε εγγραφή που βρίσκεται μέσα στο αρχείο και έχει τιμή ίση με *value* στο πεδίο με όνομα

fieldName, εκτυπώνονται τα περιεχόμενά της (συμπεριλαμβανομένου και του πεδίου-κλειδιού). Αν η παράμετρος *value* είναι NULL, τότε εκτυπώνονται όλες οι εγγραφές του αρχείου. Στο τέλος της συνάρτησης, να εκτυπώνεται επίσης το πλήθος των μπλοκ που διαβάστηκαν.
Στην περίπτωση που το value **δεν** είναι NULL, πρέπει να ψάχνετε για εγγραφές με την τιμή αυτή πραγματοποιώντας *δυαδική αναζήτηση* στο αρχείο.

int Sorted_checkSortedFile(
const char file,* */* όνομα αρχείου */*
int fieldNo */* αύξων αριθμός πεδίου */**)**

Η συνάρτηση *Sorted_checkSortedFile* ελέγχει άν το ταξινομημένο αρχείο με όνομα την τιμή που δίνεται στο όρισμα είναι ταξινομημένο με βάση το πεδιο υπ' αριθμό *fieldNo*. Επιστρέφει 1 αν είναι ταξινομημένο σωστά, 0 αν δεν είναι.

int Sorted_mergeFiles(
const char file1,* */* όνομα 1ου αρχείου προς συγχώνευση */*
const char file2,* */* όνομα 2ου αρχείου προς συγχώνευση */*
int fieldNo */* αύξων αριθμός πεδίου */**)**

Η συνάρτηση *Sorted_mergeFiles* συγχωνεύει <u>ταξινομημένα</u> (merge) τα δύο *BF* αρχεία, τα ονόματα των οποίων δίνονται ως ορίσματα. Για τη συγχώνευση τους, τα αρχεία *πρέπει* να είναι ταξινομημένα ως προς fieldNo. Ο έλεγχος αυτός πρέπει να γίνεται εσωτερικά στη συνάρτηση από εσάς. Η συνάρτηση επιστρέφει 0 σε περίπτωση επιτυχίας ή -1 σε περίπτωση σφάλματος.

Πιο συγκεκριμένα, η λειτουργικότητα που πρέπει να υλοποιηθεί είναι η ακόλουθη:

1) Να δημιουργείται ένα νέο αρχείο επιπέδου *BF*, με όνομα τη συνένωση των τιμών των ορισμάτων <file1>,<file2>,<fieldNo>. Για παράδειγμα, αν τα ορίσματα έχουν τιμές **file1 = "A", file2 = "B", fieldNo = 2** τότε το αρχείο που πρέπει να δημιουργηθεί, θα έχει όνομα **"AB2"**.
2) Να διαβάζονται εγγραφές από τα αρχεία *file1* και *file2* και να εισάγονται στο νέο αρχείο ταξινομημένες ως προς το πεδίο υπ' αριθμό *fieldNo,* αφού έχετε ελέγξει ότι το κάθε αρχείο είναι ήδη σωστά ταξινομημένο ως προς το πεδίο αυτό.

<u>Παράδοση εργασίας</u>

Η εργασία είναι ομαδική, **2 ή 3 ατόμων**.

Γλώσσα υλοποίησης: C / C++

Περιβάλλον υλοποίησης: Linux (gcc 4.3+) ή Windows (Visual Studio 2008). Η επιλογή έγινε με βάση τα διαθέσιμα εργαλεία ανάπτυξης που υπάρχουν στη σχολή.

Παραδοτέα: Τα αρχεία πηγαίου κώδικα (sources) και τα αντίστοιχα αρχεία κεφαλίδας (headers) καθώς και αρχείο readme με περιγραφή / σχόλια πάνω στην υλοποίησή σας.

Προθεσμία Παράδοσης: 2 Μαΐου 2011, 5:00 μμ