Πληροφορική & Τηλεπικοινωνίες
Υλοποίηση Συστημάτων Βάσεων Δεδομένων -
Χειμερινό Εξάμηνο 2016 – 2017
Καθηγητής Δ. Γουνόπουλος

Άσκηση 2 - Εξωτερική Ταξινόμηση
Παράδοση: 20/01/2017

Σκοπός της εργασίας αυτής είναι η κατανόηση της εσωτερικής λειτουργίας των Συστημάτων Βάσεων Δεδομένων όσον αφορά τη διαχείριση σε επίπεδο μπλοκ (block) αλλά και ως προς τη διαχείριση σε επίπεδο εγγραφών. Αφορά σε μία βασική λειτουργία που πραγματοποιείται συχνά τις βάσεις δεδομένων και είναι η *ταξινόμηση αρχείων* (sort files).
Στα πλαίσια της 2ης εργασίας θα υλοποιήσετε ένα πλήθος συναρτήσεων ώστε με είσοδο το όνομα ενός αρχείου, θα δημιουργείται ένα νέο αρχείο προορισμού, στο οποίο θα τοποθετούνται ταξινομημένες οι εγγραφές που περιείχε το αρχείο εισόδου, Η ταξινόμηση θα γίνεται ως προς το συγκεκριμένο πεδίο που θα δίνεται ως όρισμα κατά την κλήση της συνάρτησης. Η συγκεκριμένη λειτουργικότητα πρέπει να υλοποιηθεί πάνω από το επίπεδο διαχείρησης μπλοκ (block).
Οι συναρτήσεις που καλείστε να υλοποιήσετε θα έχουν το πρόθεμα Sorted_. Το αρχείο εισόδου και το ταξινομημένο αρχείο εξόδου θα έχουν μέσα εγγραφές της ***ίδιας μορφής*** και συγκεκριμένα θα έχουν τη μορφή που δίνεται στη συνέχεια.

```
typedef struct{
        int id,
        char name[15],
        char surname[20],
        char city[25];
}Record;
```

Το πρώτο μπλοκ (block) κάθε αρχείου, περιλαμβάνει “ειδική” πληροφορία σχετικά με το ίδιο το αρχείο. Η πληροφορία αυτή χρησιμεύει στο να αναγνωρίσει κανείς τί είδους είναι το αρχείο, κ.λπ.

Στη συνέχεια δίνεται ένα παράδειγμα των εγγραφών που θα περιέχονται στα αρχεία που θα διαχειρίζεστε σε αυτή την άσκηση. Εγγραφές με τις οποίες μπορείτε να ελέγξετε το πρόγραμμά σας θα δοθούν υπό μορφή αρχείων κειμένου μαζί με κατάλληλες συναρτήσεις main.

{15, “Giorgos”, “Dimopoulos”, “Ioannina”}
{4, “Antonia”, “Papadopoulou”, “Athina”}
{300, “Yannis”, “Yannakis”, “Thessaloniki”}

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 **512** bytes.

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 blockFile /* αναγνωριστικό αρχείου block */)
Η συνάρτηση BF_CloseFile κλείνει το ανοιχτό αρχείο με αναγνωριστικό αριθμό blockFile. 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 blockFile /* αναγνωριστικό αρχείου block */)
Η συνάρτηση Get_BlockCounter δέχεται ως όρισμα τον αναγνωριστικό αριθμό blockFile 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 blockFile /* αναγνωριστικό αρχείου block */)
Με τη συνάρτηση BF_AllocateBlock δεσμεύεται ένα καινούριο block για το αρχείο με αναγνωριστικό αριθμό blockFile. Το νέο block αρχικοποιείται με μηδενικά και δεσμεύεται πάντα στο τέλος του αρχείου, οπότε ο αριθμός του block είναι BF_getBlockCounter(blockFile) - 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 Sort

Στη συνέχεια περιγράφονται οι συναρτήσεις που καλείστε να υλοποιήσετε στα πλαίσια της εργασίας.

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

Η συνάρτηση *Sorted_CreateFile* χρησιμοποιείται για τη δημιουργία και κατάλληλη αρχικοποίηση ενός άδειου αρχείου σωρού 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 στο αρχείο. Επιστρέφει τον αναγνωριστικό αριθμό ανοίγματος αρχείου, όπως αυτός επιστράφηκε από το επίπεδο διαχείρισης μπλοκ ή -1 σε περίπτωση σφάλματος. Αν το αρχείο που ανοίχτηκε δεν πρόκειται για ταξινομημένο αρχείο σωρού, τότε αυτό θεωρείται επίσης 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_InsertEntry(int fileDesc, /* αναγνωριστικός αριθμός ανοίγματος αρχείου */
Record record /* δομή που προσδιορίζει την εγγραφή */)

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

void Sorted_SortFile(char *fileName, /* όνομα αρχείου */
int fieldNo /* αύξων αριθμός πεδίου προς σύγκριση */)

Η συνάρτηση αυτή ταξινομεί ένα BF αρχείο με όνομα *fileName* ως προς το πεδιο που προσδιοριζεται από το *fieldNo*. Το *fieldNo* είναι ο αύξων αριθμός του πεδίου, δηλαδή αν το *fieldNo* = 0, τότε το πεδίο ως προς το οποίο θέλουμε να κάνουμε ταξινόμηση είναι το id, αν *fieldNo* = 1, τότε το πεδίο ως προς το οποίο θέλουμε να ταξινομήσουμε είναι το name, etc. Η συνάρτηση επιστρέφει 0 σε περίπτωση επιτυχίας ή -1 σε διαφορετική περίπτωση.

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

1) Το output file θα έχει όνομα *<fileName>*,Sorted,*<fieldNo>*. Για παράδειγμα, αν τα ορίσματα ένουν τιμές *fileName* = "A", *fieldNo* = 2, τότε το αρχείο που θα δημιουργηθεί θα έχει όνομα "ASorted2".
2) Να διαβάζονται οι εγγραφές από το αρχείο *fileName* και να εισάγονται στο νέο αρχείο ταξινομημένες ως προς το πεδίο με αρίθμηση *fieldNo*. Η ταξινόμηση θα γίνει με βάση τον αλγόριθμο εξωτερικής ταξινόμησης που έχετε διδαχθεί στο μάθημα (*sort-merge*).

void Sorted_checkSortedFile(char *fileName, /* όνομα αρχείου */
int fieldNo /* αύξων αριθμός πεδίου προς σύγκριση */)

Η συνάρτηση *Sorted_checkSortedFile* ελέγχει αν το αρχείο με όνομα *fileName* είναι ταξινομημένο με βάση το fieldNo πεδίο. Επιστρέφει 0 σε περίπτωση επιτυχίας, ενώ σε διαφορετική περίπτωση -1.

void Sorted_GetAllEntries(int fileDesc, /* αναγνωριστικός αριθμός ανοίγματος αρχείου */
int* fieldNo, /* αύξων αριθμός πεδίου προς σύγκριση */
void *value, /* τιμή του πεδίου προς σύγκριση */)

Η συνάρτηση αυτή χρησιμοποιείται για την εκτύπωση όλων των εγγραφών που υπάρχουν στο αρχείο σωρού οι οποίες έχουν τιμή στο πεδίο (του οποίου ο αύξοντας αριθμός είναι *fieldNo*) ίση με *value*. Το *fileDesc* είναι ο αναγνωριστικός αριθμός ανοίγματος του αρχείου, όπως αυτός έχει επιστραφεί από το επίπεδο διαχείρισης μπλοκ. Η παράμετρος *fieldNo* μπορεί να παίρνει μία από τις εξής τιμές: 0, 1, 2, 3, που αντιστοιχούν στα πεδία μιας εγγραφής: "*id*", "*name*", "*surname*" και "*city*".

Για κάθε εγγραφή που βρίσκεται μέσα στο αρχείο και έχει τιμή *value* στο πεδίο που καθορίζεται από το *fieldNo*, να εκτυπώνονται τα περιεχόμενά της (συμπεριλαμβανομένου και του πεδίου-κλειδιού). Αν η παράμετρος value είναι NULL, τότε εκτυπώνονται όλες οι εγγραφές του αρχείου. Η συνάρτηση εκτελεί δυαδική αναζήτηση για την εύρεση των εγγραφών με τη ζητούμενη τιμή αν η παράμετρος value δεν είναι NULL. Να εκτυπώνεται επίσης το πλήθος των μπλοκ που διαβάστηκαν.

Παράδοση εργασίας

Η εργασία είναι ομαδική **2 ατόμων**.

Ημερομηνία Παράδοσης: 20/01/2017
Γλώσσα υλοποίησης: C / C++
Παραδοτέα: Τα αρχεία πηγαίου κώδικα (sources) και τα αντίστοιχα αρχεία κεφαλίδας (headers) καθώς και readme αρχείο με περιγραφή / σχόλια πάνω στην υλοποίησή σας.

Για απορίες χρησιμοποιείτε της αντίστοιχες συζητήσεις του piazza. Να παρακολουθείτε τις συζητήσεις γιατί δίνονται διευκρινίσεις από τους βοηθούς.