Database Systems Lab

SESSION 3

Building single-level persistent primary index for a data file

In this lab session, you will build a PERSISTENT single-level primary index as part of the Library system (LIBSYS) implementation. You are expected to build on the LIBSYS implementation you created from SESSION 2.

Complete the following tasks:

Modify the LIBSYS function as per the following:

// libsys_open
// Open the data file and index file in rb+ mode
// Update the fields of LIBSYS_RepoInfo appropriately
// Read index entries from the index file and save into ARRAY
// Close only the index file
int libsys_open(char *repo_name);

// libsys_load_ndx
// Internal function used by libsys_open to read index entries into array
int libsys_load_ndx();

// put_book_by_key
// Seek to the end of the data file
// Create an index entry with the current data file location using ftell
// Add index entry to array using offset returned by ftell
// Write the key at the current data file location
// Write the record after writing the key
int put_book_by_key(int key, struct Book *rec);

// get_book_by_key
// Search for index entry in index array
// Seek to the file location based on offset in index entry
// Read the key at the current file location
// Read the record after reading the key
int get_book_by_key(int key, struct Book *rec);

// libsys_close
// Open the index file in wb mode (write mode, not append mode)
// Unload the index array into the index file (overwrite the entire index file)
// Close the index file and data file
int libsys_close();

Testing

Two testing programs are given to you.

a. First test with simple_driver.c

b. Then test your program with libsys_tester.c program. This program takes a test file as input to perform automated testing with large number test cases and scenarios.

Submission

YOU ARE NOT EXPECTED CHANGE ANY OF THE FILES GIVEN TO YOU. Upload only rollno_lab3.c to LMS.

```
gcc -o output libsys.c simple_driver.c
./output
```

```
gcc -o output libsys.c libsys_tester.c
./output testcase.in
```

Persistent Indexing : Sample Output

Test case: CREATE newdemo 0
Status PASS:

Test case: OPEN newdemo 0
Status PASS:

Test case: STORE 10000 0
Status PASS:

Test case: STORE 10001 0
Status PASS:

Test case: STORE 10002 0
Status PASS:

Test case: SEARCH 10000 0
Status PASS:

Test case: SEARCH 10000 0
Status PASS:

Test case: SEARCH 90000 1
Status PASS:

Test case: CLOSE 0
Status PASS:

Test case: OPEN newdemo 0
Status PASS:

Test case: STORE 10000 0
Unable to add contact with key 10000. Error 2Status FAIL: add_contact returned status 1

Test case: STORE 10001 0
Unable to add contact with key 10001. Error 2Status FAIL: add_contact returned status 1

Test case: STORE 10002 0
Unable to add contact with key 10002. Error 2Status FAIL: add_contact returned status 1

Test case: SEARCH 10000 0
Status PASS:

Test case: SEARCH 10000 0
Status PASS:

Test case: SEARCH 90000 1
Status PASS:

Test case: CLOSE 0

Status PASS:

Test case: OPEN newdemo 0
Status PASS:

Test case: STORE 10003 0
Status PASS:

Test case: CLOSE 0
Status PASS:

Test case: OPEN newdemo 0
Status PASS:

Test case: SEARCH 10003 0
Status PASS:

Test case: SEARCH 10000 0
Status PASS:

Test case: SEARCH 90000 1
Status PASS:

Test case: CLOSE 0
Status PASS: