|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **National University of Computer and Emerging Sciences, Lahore Campus** | | | | |
| final design | **Course:** | **Operating Systems** | **Course Code:** | **CL205** |
| **Program:** | **BS(Computer Science)** | **Semester:** | **Spring 2018** |
| **Duration:** | **150 Minutes** | **Total Marks:** | **100** |
| **Paper Date:** | **10-5-2018** | **Weight** | **40%** |
| **Section:** | **D & F** | **Page(s):** | **2** |
| **Exam:** | **Lab Final** | **Roll No:** |  |

|  |  |
| --- | --- |
| **Instructions/Note:** | 1. You must ensure that you have made proper submission of your code. No Issues will be entertained later on. 2. Discussion with other students is not allowed. 3. Use of the internet, notes, codes, lab manuals, and flash drives is strictly prohibited. You can though refer to man pages. 4. Plagiarism will result in F grade in lab. 5. No previous code of yours or lab manuals are not allowed. 6. Submissions are on slate   7. Submission file’s name should be your roll no and question no L1X-XXXX\_QX.cpp |

You have to write a client-server model where a server stores data in a file books.dat. The clients can request the server to retrieve data from books.dat or make changes to books.dat. You have to use sockets to communicate between the server and the client.

The server has a file books.dat, which stores the following attributes related to a book: the title, the author’s name, publishing year and the ISBN. You can use the following structure for storing the books attributes on the server side.

Struct record {

char title[25];

char aut\_name[25];

char ISBN [25];

char pub\_year[25];};

**The multithreaded server** – As soon as a new client connects to the server, the server will create a thread to service the new client. The multithreaded server will handle a maximum of 5 clients at a time. For efficiency, the server will have the file - books.dat - mmap-ed for accessing/modifying. Because multiple clients can request the server, you have to take care of the synchronization or/and mutual exclusion.

**The client** – the clients on the other hand can request the server to

1. Add a book record to the file
2. Modify a specific attribute in a particular record in books.dat
3. Delete a record
4. Search for a particular record

After connecting to the server, the client will send one of the following strings to the server: “query”, ”add”, “modify” or “delete”.

The “query” operation – will let the client(s) search for a specific book in books.dat with respect to the **title only**. After the server has received a request for querying, the server will wait for the client to pass a book title next. If a record with that particular title exists in books.dat, the client will receive four tokens (strings/ book attributes) from the server, other wise a string “Not Found.” is sent.

The “add” operation – will allow the client(s) to add a record in books.dat. After the server has received a request for adding a record, the server will stand by to receive the four attributes: the title, the author’s name, publishing year and ISBN. The four attributes will be added to any free location (as a record) in books.dat. A free location in books.dat will contain whitespaces equal to the size of the struct record.

The “delete” operation – will allow the client(s) to delete a record from the books.dat. After the server has received a request to delete, the server will stand by to receive the title of the book. The server will delete the record with the particular title in books.dat by inserting whitespaces equal to the size of the struct record in the location of that particular record.

The “modify” operation – will allow the clients to modify a particular record in books.dat. After the server has received a request to modify, the server will stand by to receive the title of the record to modify, and the four tokens (strings/book attributes). The server will overwrite the old record with the newly received record.

The add, delete and modify operations will return a string “Success!” if the operations complete successfully. Otherwise a string “Error!” is sent back to the client.

Note:

You can assume that the tokens are always less than 25 bytes in size.

The file books.dat has the capability to store 10 records.

And the tokes are always received in the following order: title, author’s name, ISBN and publishing year.

The following window shows a sample output for the add operation.

|  |  |
| --- | --- |
| $ ./server  add request received  standing by to receive tokens  Tokens received  Operating Systems  Silberchartz  1234567890  1995  Added Successfully | $ ./client  Connected.  add  Operating Systems  Silberchartz  1234567890  1995  Success.  Exiting client.  $ |