Com S 454/554 Fall 2020 Assignment 2

Requirement

Requirement: Development of a Client/Server-based Integer Storage System

In this assignment, you are required to develop a distributed integer storage system following the client/server architecture by utilizing RPC as communication middleware.

The server should maintain an ordered list of integers (initially empty) in memory, and provide the following procedures (functions) to the clients:

- append(argument: an array of 10 integers) – append the integers at the end of the ordered list;
- query(argument: "pos" which is an integer) – return the integer currently on position "pos" in the ordered list (note: the positions start with 0 and remain contiguous);
- remove(argument: "pos" which is an integer) – delete the integer on position "pos" in the ordered list and the remaining integers may be shifted to keep the positions contiguous.

With accesses to the above functions, each client should provide the following user interface:

- when a client is launched, it accepts the following inputs as command-line arguments:
  - hostname or IP-address of the server;
  - type of request, which is a lower-case string of value "append", "query" or "remove";
  - the parameters
    - for "append", 10 integers that should be appended to the ordered list;
    - for "query", 1 integer specifying the position of the integer to be queried;
    - for "remove", 1 integer specifying the position of the integer to be removed;
  - multiple parameters are separated by a blank;
  - when a function can't be performed, print out appropriate error message (for example, query/remove position exceeds the scope of the ordered list).

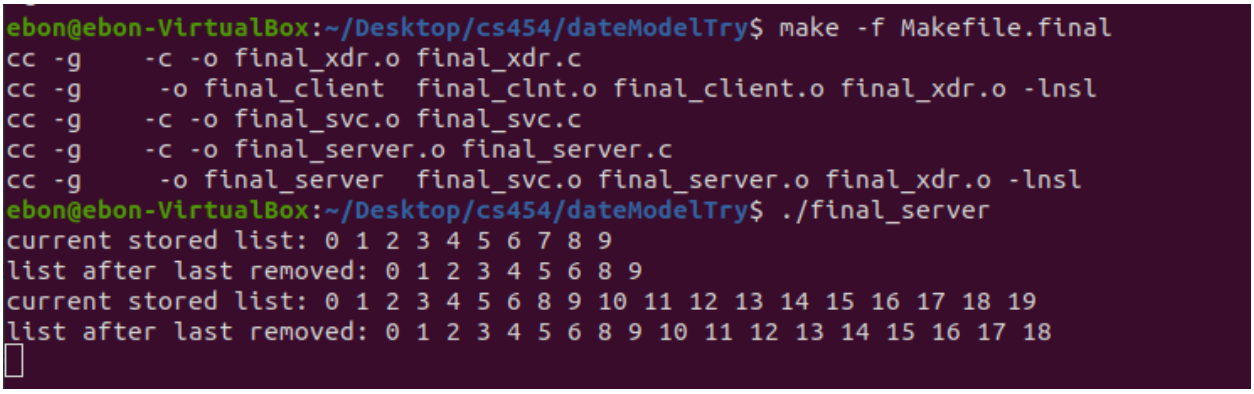
Environment setup and general process

1. Linux environment:
   a. Putty: SSH: pyrite.cs.iastate.edu (permission issue ×), connects to school U drive.
   b. Ubuntu on windows (permission issue and package issue ×)
   c. Google ISU Linux, SSH putty and cd /myfiles/<username> to connect to school U drive.
   d. Ubuntu on VirtualBox √
   (https://www.youtube.com/watch?v=QbmRXJJKsvs&t=283s)
2. Install rpcgen, rpcbind
3. Write and compile your IDL file. .x file format
   a. IDL: interface Difinition Language
4. "rpcgen -a -C filename.x"
5. get filename_server.c, filename_client.c. implement those two files.

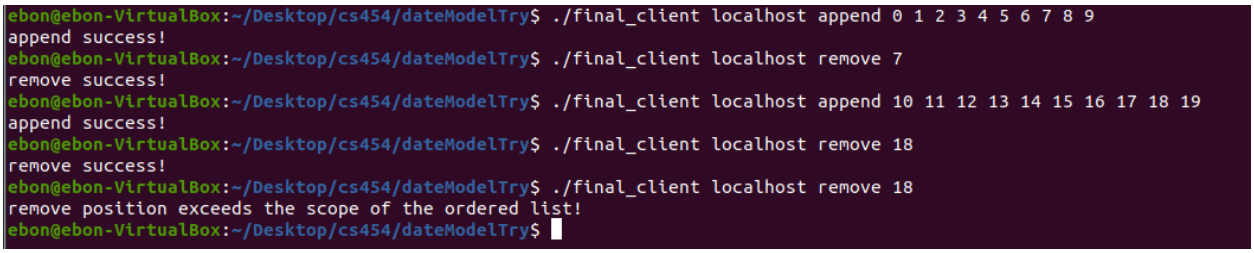
6. “make -f Makefile.filename” to compile all c code.
7. How to run:
    a. Run server file first: ./server
    b. Run client file: ./client localhost append + 10 integers
    c. Run client file: ./client localhost query + 1 integer
    d. Run client file: ./client localhost remove + 1 integer
8. Note:
    - When declare typedef int arr[10]; in x file, remember to change from
      int append_1_arg;
      to
      *int *append_1_arg;*
      in client.c file.
    - Global value usage

Screenshot

Server side:

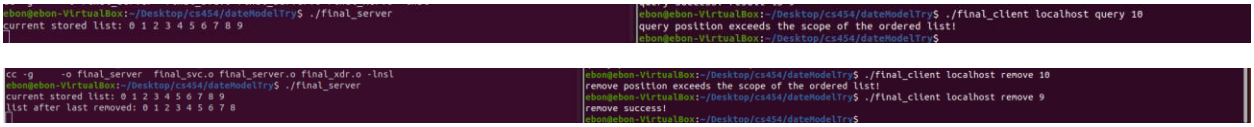


Client side:



Inappropriate input cases:

Query/remove out of bound:



Append number of values is not 10: