ELEC 477

Kevin Yu: 20203451

Raatik Sharma:

Ainsley Taylor: 20210012

2024-02-05

For example: "File test1.txt
contains the output of our first test. It shows that the value 'Smith Engineering is great" was stored with the key 1, and successfully retrieved using the key 1". You will need to have appropriate print statement in the client so that the output shows that the test is successful.

Test 1

1. *The first test should have a simple client that puts a value and retrieves it and checks that it got the same value back.*

Test 1 asks us to perform a PUT and GET RPC. In our client.cpp file we called a PUT request with a value of “kevin/0yu” with a key of 7. Test1.txt shows the output of our results when we do a GET request as shown below. This illustrates that the RPC methods for PUT and GET work correctly since we fetch the correct value that was associated with the key 7.

```
--------------- E1SERVICE: Get Request Finished ---------------
Client's GET status: 1
Client's GET value: kevinyu
Client's GET length: 8
Main: **********************************
Main: calling stop services on server
Main: **********************************
Main: waiting for threads to complete
Main: shutting down protobuf library
```

Test 2

2. You must test that data containing null bytes is stored and retrieved correctly.

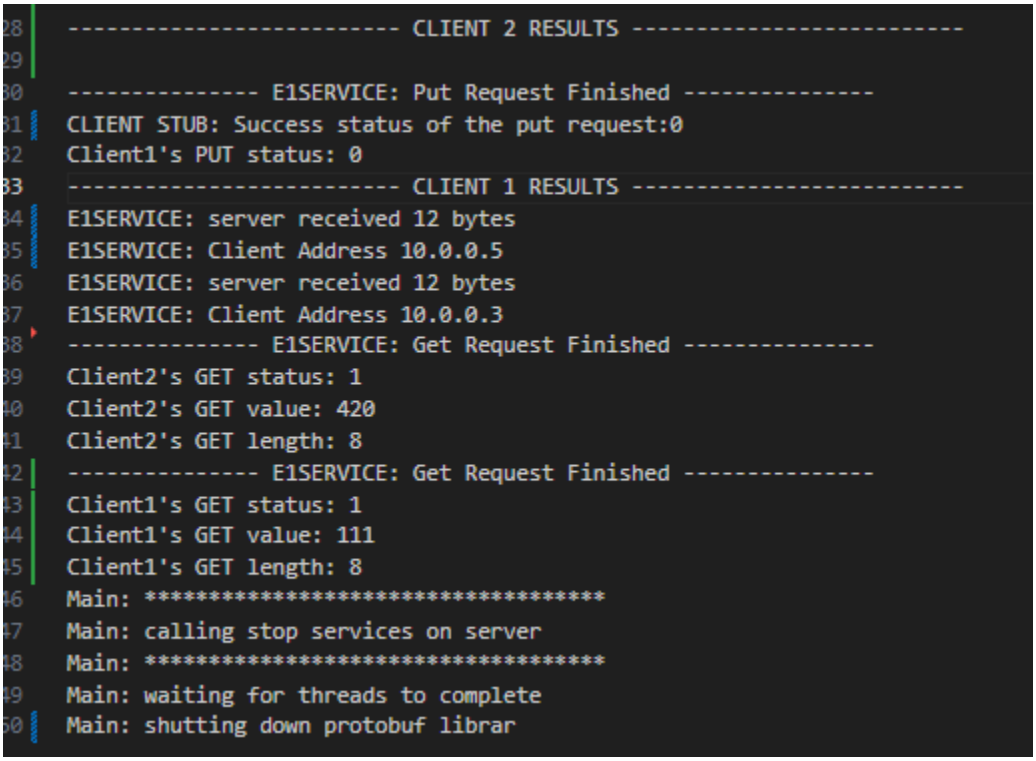
Test 2 asks us to see if our RPC works with storing and returning null bytes. In this test we do the same PUT and GET request as in Test 1. But we illustrate that storing and retrieving null bytes works correctly since “kevin\0yu” contains a null byte in the middle indicating that the end of the string is after the name “kevin” and “yu” would not be stored theoretically. However, we see from the image below that our GET request does indeed return back “kevinyu” meaning our RPC method can store and retrieve null bytes.

```
--------------- E1SERVICE: Get Request Finished ---------------
Client's GET status: 1
Client's GET value: kevinyu
Client's GET length: 8
Main: **********************************
Main: calling stop services on server
Main: **********************************
Main: waiting for threads to complete
Main: shutting down protobuf library
```

Test 3

3. You should have tests with two servers and two clients to show independence.

In Test 3, we setup two independent clients and two independent servers each with their unique addresses and each server has its own unique database. We call a PUT request for each client with the exact same key “69” but client 1 has a value of “111” and client 2 has a value of “420”. When we call the GET request on each client with the same key of “69” we see that each client returns their respective value that they stored with their PUT request. This illustrates that each client and server pair are independent of one another.



Test 4

4. You should have tests that write different values to the same server with the same key

to overwrite, this test should be timed so that if one of the packets is delayed, the

value returned at the end by one of the clients is different.

In Test 4, we have two clients that write two PUT requests and then call two GET requests. However, the first client’s GET request is delayed meaning that the PUT request from the second client will overwrite the value in the gdbm database. When both clients return their values, we see that the GET responses print out the second client’s stored value. Therefore the second client has overwritten the first client’s initial value.

```
--------------- E1SERVICE: Put Request Finished ---------------
CLIENT STUB: Success status of the put request:0
------------------------- CLIENT 2 RESULTS -------------------------
E1SERVICE: server received 12 bytes
E1SERVICE: Client Address 10.0.0.5
--------------- E1SERVICE: Get Request Finished ---------------
Client2's GET status: 1
Client2's GET value: 586
Client2's GET length: 8
------------------------- CLIENT 1 RESULTS -------------------------
E1SERVICE: server received 12 bytes
E1SERVICE: Client Address 10.0.0.3
--------------- E1SERVICE: Get Request Finished ---------------
Client1's GET status: 1
Client1's GET value: 586
Client1's GET length: 8
Main: **********************************
Main: calling stop services on server
Main: **********************************
Main: waiting for threads to complete
Main: shutting down protobuf library
```