

CS/CE/TE 6378: Project I

Instructor: Ravi Prakash

Assigned on: February 9, 2016

Due date and time: February 28, 2016, 11:59 pm

This is an individual project and you are expected to demonstrate its operation to the instructor and/or the TA. Sharing of code among students, or reusing code from previous semesters or from other sources without the permission of the instructor is prohibited. Any violation of this policy can result in the student(s) being reported to appropriate authorities for academic misconduct.

1 Requirements

1. Source code must be in the C /C++ /Java programming language.
2. The program must run on UTD lab machines (dc01, dc02, . . . , dc45).

2 Client-Server Model

In this project, you are expected to use *client-server* model of computing. Knowledge of threads and socket programming and its APIs in the language you choose is expected. Each process (server/client) must execute on a separate machine (dcxx).

3 Description

Implement a solution that mimics a banking system where the bank maintains three identical files at three different sites, maintaining account information. A client, executing at a fourth site, may issue read or write requests. All replicas of a file are consistent, to begin with. The desired operations, as far as reading and writing are concerned, are as follows:

- a. The client may send a `read from file REQUEST` to any of the sites (randomly selected). In this case, the site should reply with requested content (if available).
- b. The client may send a `write to file REQUEST` to any of the sites (randomly selected). In this case, the site must report a `successful` message to the client if *all* copies of the file, across the three sites, are updated consistently. Otherwise, the site must report a `failure` message to the client.

As part of this project you have to determine how a site, on receiving the `write to file REQUEST` communicates with the other sites to get the same write performed on all the copies. Your program must support the creation of new files, writes to the end of files, reads and writes at specific offsets from file beginning. It must also report an error if an attempt is made to read from a file that does not exist at the requested site. An error must be reported if there is an attempt to write to a file that does not exist on all the sites.

Assume there is only one client. When the client starts it must present the user with five options: create a new file, move to a specific location in a file that has already been opened, issue a read request, issue a write request, or end the session. If the user chooses to create a new file, move to a certain offset, or perform a read or write, then the client should obtain the necessary information, initiate the action, return the result, and go back to presenting the five options mentioned above. A client issues only one request at a time and blocks until the response to that request is received. So, for this project you need not worry about concurrent read/write requests for the same object. Your client should gracefully terminate when the user chooses to end the session.

The teaching assistant will upload a trace file listing the a sequence in which the five commands are to be executed, along with their parameters. Make sure that your code correctly executes all the commands. However, your code should be able to correctly execute other sequences of these commands for you to get credit for this project.

4 Submission Information

The submission should be through eLearning in the form of an archive consisting of:

1. File(s) containing the source code.
2. The README file, which describes how to run your program.

DO NOT submit unnecessary files.