This README file explains how to run a multi threaded client server application where server is set on cloud and multiple requests from the multiple clients are passed through TCP connection using socket programming to a predefined port. To avoid the discrepancies caused by the simultaneous access of same shared file from multiple clients, we used locking mechanism in the server side. And also to improve the performance of the application, each thread is assigned to each specific core in CPU. In this, time stamps of requests/responses (when the request is made and response is received ) are provided from both client side and server side.

**Getting Started**

Unzip the FileServerDelivarables.zip

In terminal cd to the location where unzipped file is downloaded

Run the command *java -jar fsClient.jar*

**Inputs**

There are three operations as mentioned below . Console prompts to

1. Enter the operation (case-insensitive)i.e.,

R == READ

W == WRITE

D == DELETE

1. Enter the filename ( if file does not exist, a new file will be created in current directory in case of write operation and in rest of the operations, IO Exception occurs)
2. In case of only write operation, application suggests to Enter the message that has to be appended to the working file.

**Running the tests**

Run the following test cases to check locking mechanism and multi threading process.

1. Multiple runs of fsClient.jar file to initiate multiple same operations on same file
2. Multiple runs of fsClient.jar file to initiate multiple different operations on same file
3. Multiple runs of fsClient.jar file to initiate multiple same operations on different file
4. Multiple runs of fsClient.jar file to initiate multiple different operations on different file

**Built With**

JRE: 1.8.0\_112-release-736-b21 x86\_64

Mac OS X 10.13.1/Microsoft Windows Data Server 2016 Datacenter

CPU: Intel Xeon CPU E5-2676 @2.4GHz, 1 Core(s)

Following is required while setting up the server

Amazon EC2: t2.micro instance qualifying free tier.