Jinghao Shen
StudentID: 1579961

I first test the connection between server and client with user defined ip address and port number.
Then, math functions.
Client requests with one argument add or sub or mul without overflow.
Client requests with one argument add or sub or mul with overflow.
Client request with more than one argument.
Client sends requests byte by byte.
Then file create and file size functions.
Create a new file.
Create an existing file.
Return the file size of Makefile
Return the file size of a non-existent file
For the read and write functions.
Read from a file 100 bytes
Read from a file 100 bytes with an offset 2
Write to a file 5 bytes
Write to a file 5 bytes with an offset

About what fraction of your design and code are there to handle errors properly? How much of your time was spent ensuring that the server behaves “reasonably” in the face of errors?
In math function, 80% is for error handling. For other parts of the program error handling is about 20%. About only 15% of my time was spent on error handling.

What happens in your implementation if, during an RPC call, the connection is closed before the message is finished? How could this happen in the first place?
There’s a timer, if it passes a value, the connection to the client will close and wait for another client. Server won’t close.
The client can close connections at any time or the data loss in transmitting.