## Assignment1—Report

Code of client:



Create a stub which is a proxy for RMI server on client side and handles remote invocation on behave of RMI client. And try to lookup on the Registry Service.

Code in put() method of client:



In this snippet, if the mode is passive which means the server should start a thread to waiting for the client send socket connection, in here, we remotely invoke the method svr.put() to start the thread in putThread() for the server side. The client build the socket and get connected to the server, create the dataoutputstream object to handle the file bytes and send it to the server .After finished the while loop print out the sentence “transfer file success” and close those object; if the mode is active which indicate that the client will run a putThread() to waiting for the server to connect. At the same, invoking the method of Iserver to make such connection.

Code in the get() method of client:



In this get() method, client using this to get file from server side, when the mode is PASSIVE,

First call the get method of the server which will start a thread to listen. The client start a socket object xfer to connect which is quite same with the code above ,just write and read from the buffer and input,output stream. If the mode is active, the client just start the getThread and waiting for the server to make the connection. Invoking the get()method to let the server know what is going on.

Code in getThread() method of client:



This method will be invoked when the mode is ACTIVE and the command is get the file from the server. First ,getting the file name and file size for printing ,and start read the bytes from the DataInputStream object.Using the FileOutputStream to write the bytes into file.

Code in putThread() method of client：



In this piece of code, it is quite similar to the GetThread() method ,just start a new thread to waiting for the server to connect. And write the data into the DataOutputStream object.

Now it is time of the server side

The code in put() for the server:

Server will run the code when the client remotely invoke, if the mode is ACTIVE, the server will start the connection to the client with sending the file name and file size. If the mode is passive, just call the GetThead Methods to start a new thread to wait for the connection.

The put method of server side：



If the put method is called and mode is equal to ACTIVE, the server start connecting to client and begin to receive file from the client and output the file in the current path. When the mode is PASSIVE, the server just start a new thread to listen the connection from the clients.

The getThread of server:



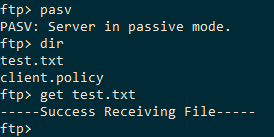
The PutThread of server:



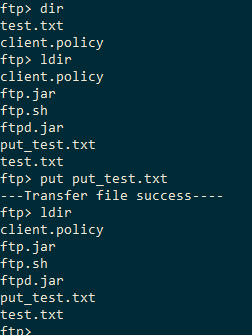
Testing code: Start the server in local and start client, when the client found the proxy server in the registry, the command line start to show.

Test in the passive mode:

The get method:



The put method:

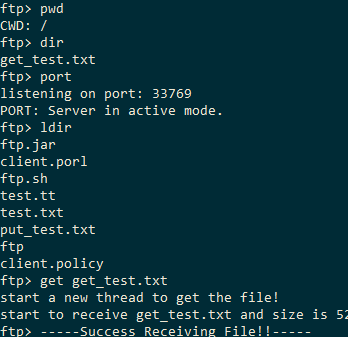


Testing in the passive mode, to get or put file with the server and correct the code, make sure the file that been transferred is complete.

Changing it to active mode and do the same thing like the get and put methods.

Using dir to show the current root that the get\_test.txt is already in the server side:

Test the get method：



Test in active mode:

Using the put methods to put the put\_test.txt:



The video will make sure the code is right and show the demon.