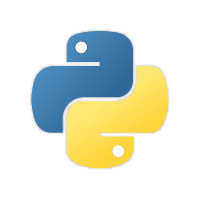
Socket Programming

Assignment 1

CSC/CPE 138

Computer Network and Internet

Professor Jun Dai



Vaukee Lee

Introduction:

In this lab, we will be putting into practice socket programming using two types of protocols which are TCP(Transmission Control Protocol) and UDP(User Datagram Protocol) in the Python programming language.

TCP in Python:

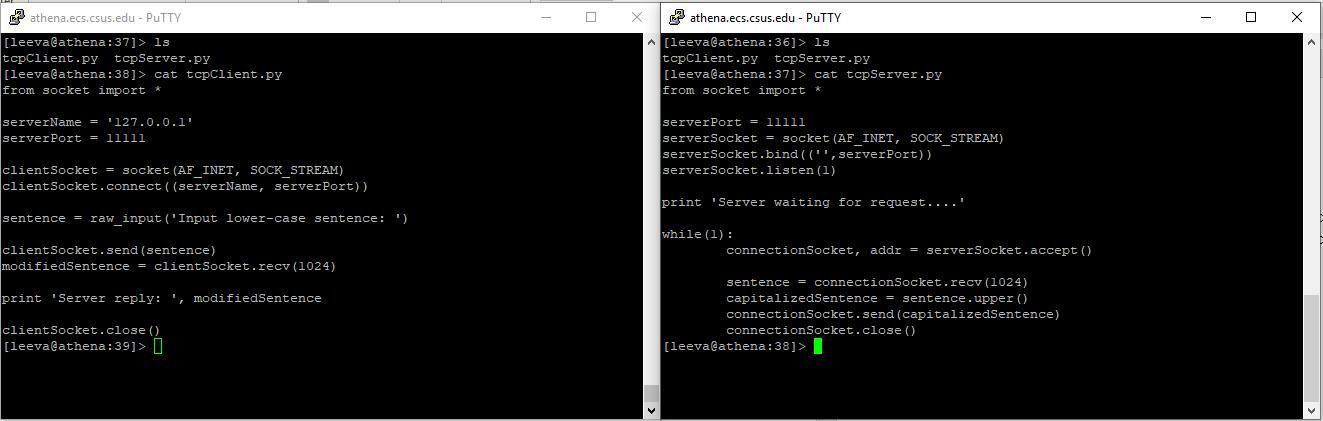


Figure 1. We have the code that was used from the example shown in the class lecture notes with some slight tweaking of course. In my code, I used a server port number of 11111 and I adjusted the outputs slightly.



Figure 2. This figure shows the Python program being ran. Once we type in a short sentence and hit enter, we get the sentence back all capitalized, but the server remains running because of the infinite loop.

UDP in Python:

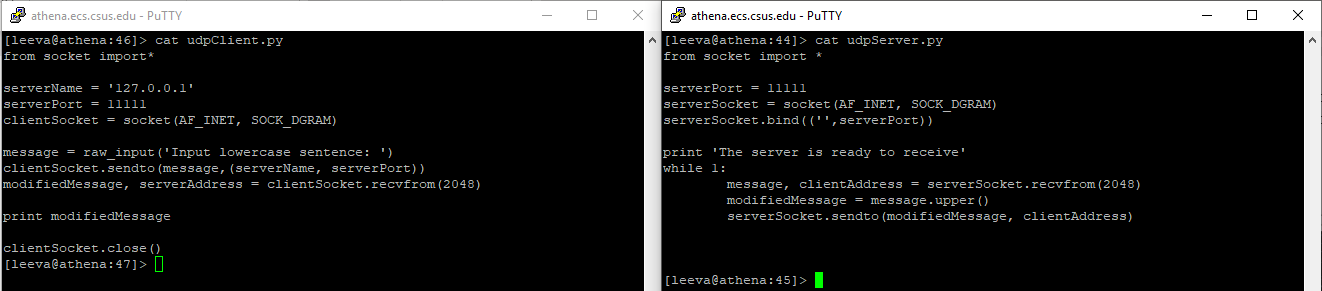


Figure 3. For this UDP part, I chose to keep the server port the same. One will be able to notice that the code is slightly different from the TCP version.

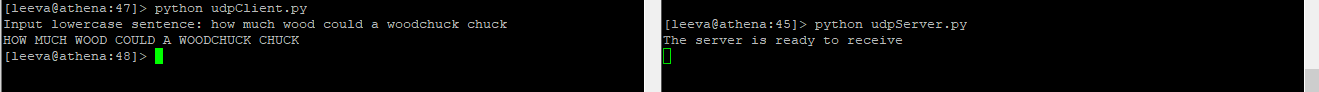


Figure 4. Like the TCP version, we also made it so that the server returns an all capitalized sentence upon the input of an all lower-cased sentence.

Conclusion:

Overall, I think this lab is a cool lab in which I would probably like to do over again in a different project. When writing up the code that was in the example, I noticed that there were key words that each protocol only used. In each of the sockets UDP/TCP, we can see the word “SOCK\_” followed by another “word.” The first one being “STREAM” for the use of a TCP connection and the second one was “DGRAM” for use with the UDP connection. Each of the sends were also different. The TCP used “send” and UDP used “sendto.” In TCP, it had an extra line of code that was called “connect,” that UDP doesn’t use signifying the trait of TCP being that it establishes a link between the client and the server.