CS 176A Report

Michael Zhang

April 12, 2020

1 Installation Prerequisites

Version: Python 3.7.7

Libraries/Frameworks: unittest, mock, sys, socket, time, subprocess

2 Usage

To run the program itself, using command-line[1], enter

```
python server_python_tcp.py [port number]
```

The server must be running before attempting to start the client. Then, on a different terminal, enter

```
python client_python_tcp.py
```

to start the TCP client. It will prompt you for an IP address, port number, and command. Assuming the three inputs are valid, it will execute the program and a file named `output.txt` will be created on the server side while one named `[command].txt` will be created on the client-side (where [command] is the command you inputted). You can verify that the file transmission completed successfully when the contents and properties of `output.txt` are the exact same as `[command.txt]` aside from the name of the file.

Similarly, the UDP server/client are run in the same manner by replacing tcp with udp in the commands above.

Note that the port number is passed as a command-line argument for servers; there exists the possibility that the port is already in use. Then, an OSError will be thrown which will be caught and our program will print "*Port in use*" and terminate.

3 Testing

I have included 4 unit-test files under the names as specified. Each can be run by using the command

```
python test_client_udp.py
```

where you replace `test_client_udp.py` with whichever file you wish to test. Passing test cases will have no Errors (as expected), and depending on the case being tested, the desired messages will be printed on console. For instance, in a test case where the socket fails to connect to the given IP and port, "Could not connect to server." would be printed and the unittest would pass as the function correctly prints out what it should.

As I use Mock, each of the unittests can be ran separately - I mock sockets depending on what is needed in my tests. I have documented my code fairly thoroughly both in unit testing and the program code itself so that it's clear which test cases are being tested and the purpose of each snippet of code. Furthermore, my program code has been modularized for easy readability.

For manual testing, you can follow the **Usage** instructions above. In any case, our program functions as intended.

[1] Note that you may need to use the command python3 instead of python as I have an alias on my configuration

4 Contributions

Thanks to Atefah/Shiyu for all the support & guidance from Piazza, StackOverflow for the countless threads I read to debug/implement my code, and GeeksForGeeks for thorough explanations!