Paul Kotys, pjk151
Philip Okoh, pco23
Internet Technology CS352
Monday, February 21, 2022

FINAL REPORT PROJECT 2
Asynchronous Sockets
TESTED ON ilab1, ilab4, cd, USING Python3

IMPORTANT: We used the features of Python3 in this assignment

QUESTION 1:
Paul Kotys pjk151, Philip Okoh pco23

QUESTION 2:
We collaborated on the project by breaking up who programmed each of the four programs: Paul implemented client.py and rs.py and Philip implemented ts1.py and ts2.py. We used GitHub and Google Drive to store our code and report, respectively.

As with the last project, most of the information we needed came from https://docs.python.org/3/library/socket.html, the Python documentation for sockets. We also used Python's threading api: https://docs.python.org/3/library/threading.html. We asked the professor and TA questions about implementation details.

QUESTION 3:
The RS creates two new threads that connect and query TS1 and TS2. The sockets that connect to TS1 and TS2 have a five second timeout, thus preventing the reads from blocking indefinitely. The two threads write their responses to global variables. The RS then joins the threads and reads the responses from the server. If both responses are empty, the RS sends the DomainName - TIMED OUT message, otherwise RS forwards the results to the client.

QUESTION 4:
All portions of the code work as required.

QUESTION 5:
Generally, we did not have any major difficulties implementing the programs. The only difficult part was testing the programs on four separate computers to make sure that everything worked correctly.

QUESTION 6:

This project required a lot of precise timing and exception handling to make the sockets work correctly across four separate programs. We learned that closing every socket with calls to shutdown(socket.SHUT_RDWR) and close() is necessary to fully terminate the connection and clean up system resources associated with the socket. We also had to learn about the Python threading api to correctly implement the simultaneous connections with TS1 and TS2.