Paul Kotys, pjk151
Philip Okoh, pco23
Internet Technology CS352
Sunday, January 30, 2022

FINAL REPORT PROJECT 1
WARMUP: SOCKET PROGRAMMING 101
TESTED ON ilab4.cs.rutgers.edu USING Python3

IMPORTANT: We used the python3 command to run client.py and server.py

QUESTION 1:
Paul Kotys pjk151, Philip Okoh pco23

QUESTION 2:
Philip and I collaborated on all aspects of project 1. Specifically, Philip wrote the initial client.py and server.py. I went through the code and fixed a few bugs related to splitting and reversing lines and timing the socket connection. We also collaborated on the report using Google Drive. We consulted the professor when we had questions.

We used https://docs.python.org/3/library/socket.html for references on Python sockets and https://docs.python.org/3/library/functions.html#open for how the open function works. We also used pdb to debug the code: https://docs.python.org/3/library/pdb.html. Finally, to figure out why the socket wasn't timing out, we found this very informative blog post: https://medium.com/pipedrive-engineering/socket-timeout-an-important-but-not-simple-issue-with-python-4bb3c58386b4

QUESTION 3:
When removing the sleep function calls, we got the following error and traceback:

```
[S]: Server socket created
[C]: Client socket created
Exception in thread server:
Done.
Traceback (most recent call last):
  File "/usr/lib/python3.8/threading.py", line 932, in _bootstrap_inner
        self.run()
  File "/usr/lib/python3.8/threading.py", line 870, in run
        self._target(*self._args, **self._kwargs)
  File "proj.py", line 18, in server
```

```
        ss.bind(server_binding)
OSError: [Errno 98] Address already in use
Exception in thread client:
Traceback (most recent call last):
  File "/usr/lib/python3.8/threading.py", line 932, in _bootstrap_inner
        self.run()
  File "/usr/lib/python3.8/threading.py", line 870, in run
        self._target(*self._args, **self._kwargs)
  File "proj.py", line 48, in client
        cs.connect(server_binding)
ConnectionRefusedError: [Errno 111] Connection refused
```

This error occurred because the server thread socket couldn't bind to the ip+port and the client thread socket couldn't connect to the ip+port. The reason this happened is because by removing the sleep calls, we didn't allow the client and server threads to properly close the socket in the previous iteration of the program. When we run the program again, the OS says that the port is already in use (causing the server thread to not bind), and the client thread can't connect since there is no program listening on that port.

QUESTION 4:
Our program works exactly as the description specifies.

QUESTION 5:
The only difficulties were thinking about edge cases. For instance, our code needs to handle the case where the input file is not terminated with a newline. We also need to handle the case where the input file has blank lines or tabs. The most difficult problem we ran into was having to set a timeout on the socket so that recv would throw an exception after a certain amount of time. This is necessary since the connection might close while recv or send is working.

QUESTION 6:
This project taught us that exceptions are a very useful tool in the Python language to facilitate socket programming. Exceptions are much easier to work with than the numeric error codes that the C socket interface uses. For instance, our server code used .settimeout() to set a timeout on the server socket operations. After five seconds, a blocking call to recv or send will raise a timeout exception, which will be caught in an except clause that terminates the while loop. This information can be found in https://docs.python.org/3/library/socket.html. The server loop will only terminate when an exception (such as timeout) occurs, or when reading from the socket returns an empty string "".