

Project Report

Project Track - Socket Programming

Team Members:

- 1.) Ayush Sawlani - IMT2018014
- 2.) Aravind Veluri - IMT2018010
- 3.) Ishmeet Singh Saggu - IMT2018030

General Outline:

Our project is on creation of a simple chat server, to which a client socket is to be connected and then the conversation between the client and server begins. Both the client and server can send and receive messages upto length of 1KB simultaneously. Client and server both can send files of any type to the other in chunks of size 1KB. Empty messages and files that do not exist in the directory of server and client programs cannot be sent from server and client respectively. Client can quit the conversation by sending "quit()" message to the server. The server and client programs can be run from same directory or from different directory in same devices and even from other devices.

Steps:

- 1.) First, we created a socket in the server file, of address family IPv4, which communicates through TCP internet protocol, and binded it to an IP address(or empty string to accept connections from any device on the same network) of server through a given port and let it listen to and accept connection requests.
- 2.) Next, from the client file, we created a socket with same address family and protocol, and connected it to the server socket(by entering same IP address and through same port).
- 3.) Now, after the connection is established, sending and receiving functions are implemented in both the programs.
- 4.) Two threads, one for sending function and one for receiving function are created and started in order to enable sending and receiving of multiple messages simultaneously from either side. Daemon threads are used to implement the "quit()" part of the program.

- 5.) Sender and Receiver functions are modified using file handling to enable the transfer of files(in bytes) from either side and creating a new file with the filename "new_<name of sent file>".
- 6.) Colorama module is installed and imported and is used to format text and background in the terminal.

Team Work:

- 1.) Ishmeet - Ishmeet learnt the basics of socket programming and went through the socket documentation and other basic socket tutorials to understand the concepts. He made the basic socket creation, binding, connecting in both server and client files and enabled basic "Hello" message from server to client.
- 2.) Aravind - Aravind learnt sending, receiving from the sockets and closing them and created two functions for sending and receiving messages in both the files. He went through threading documentation and other threading tutorials to understand threading concept. He created sending and receiving threads for corresponding functions in both files.
- 3.) Ayush - Ayush learnt file handling in python from tutorials and references mentioned. Then he modified the sending and receiving functions in both the files and enabled file transfer between them. Any type of file can be sent and received as he read and wrote the files in bytes and in chunks of size 1KB. He learnt and installed colorama module and imported it to format text and background colours in the terminal.

References:

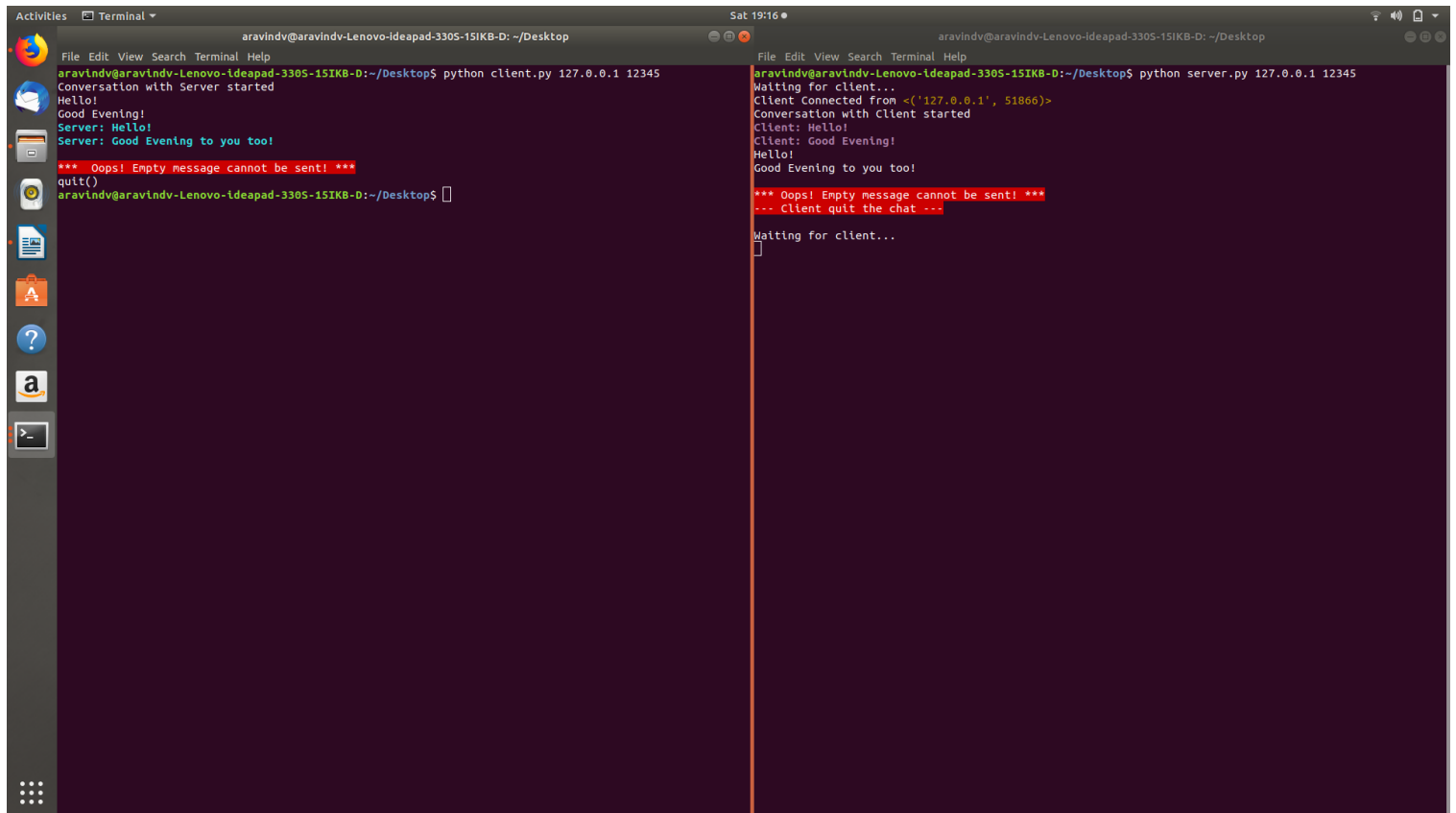
- 1.) Official socket module documentation.
<https://docs.python.org/2/library/socket.html>
- 2.) Official socket HOWTO provided in project tracks file.
<https://docs.python.org/2/howto/sockets.html>
- 3.) Official threading module documentation.
<https://docs.python.org/2/library/threading.html>
- 4.) Geeks for Geeks tutorial
<https://www.geeksforgeeks.org/socket-programming-python/>
- 5.) Threading tutorial from
<https://nitratine.net/blog/post/python-threading-basics/>
- 6.) Threading and Daemon threads tutorial from
<https://www.youtube.com/watch?v=PJ4t2U15ACo>
- 7.) File handling tutorial from
<https://www.geeksforgeeks.org/file-handling-python/>

8.) File handling in bytes tutorial from

<https://www.youtube.com/watch?v=ghLiGHpmhgg>

9.) Colorama module installed from - <https://pypi.org/project/colorama/>

Working:



```
aravindv@aravindv-Lenovo-Ideapad-330S-15IKB-D: ~/Desktop
aravindv@aravindv-Lenovo-Ideapad-330S-15IKB-D:~/Desktop$ python client.py 127.0.0.1 12345
Conversation with Server started
Hello!
Good Evening!
Server: Hello!
Server: Good Evening to you too!
*** Oops! Empty message cannot be sent! ***
quit()
aravindv@aravindv-Lenovo-Ideapad-330S-15IKB-D:~/Desktop$
```

```
aravindv@aravindv-Lenovo-Ideapad-330S-15IKB-D: ~/Desktop
aravindv@aravindv-Lenovo-Ideapad-330S-15IKB-D:~/Desktop$ python server.py 127.0.0.1 12345
Waiting for client...
Client Connected from <('127.0.0.1', 51866)>
Conversation with client started
Client: Hello!
Client: Good Evening!
Hello!
Good Evening to you too!
*** Oops! Empty message cannot be sent! ***
--- Client quit the chat ---
Waiting for client...

```

Conclusion:

We created a working server to client communication chat room with a file sending feature. We learnt basics of computer networking, socket programming in python, communication between the sockets, concept of multithreading in python, file handling in python, team work and more ideas about improving the project. Overall, we were enthusiastic to work on real world application of programming and we learnt a great deal from it and we want to improve it further.

--- THANK YOU ---