

- **Strategy to solve the problem:**
  - Before I could start programming, I had to set up my Client and Server directories.
  - Then set up the Server and Client on the CORE tool.
  - Then you for the programming part you first need to create the sockets in both the server and the client side.
  - Then you need to figure out the way to establish a connection between the client, and you do this by providing the Server IP address to the client, and you also need to make sure to provide the same port for both.
  - Then you need to ask for the name of the file on the client side and send that information to the server side.
  - And finally, the server side will send the file requested to the client side.
  - After that the only thing left to do is run the program using the setup from the start.
- **Evidence and multiple execution samples (i.e., sending different kinds of files):**

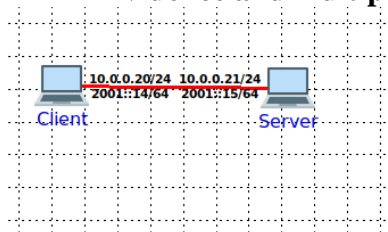


Figure 1. Figure 1 shows the topology of the connection between Client and Server running on the CORE.

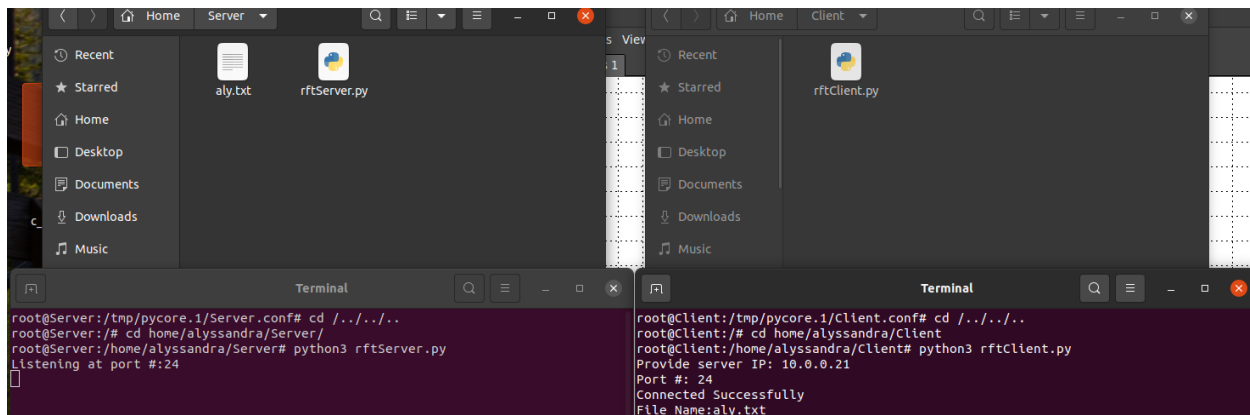


Figure 2. Figure 2 shows the program asking to the user details of the connection and the name of the file to send.

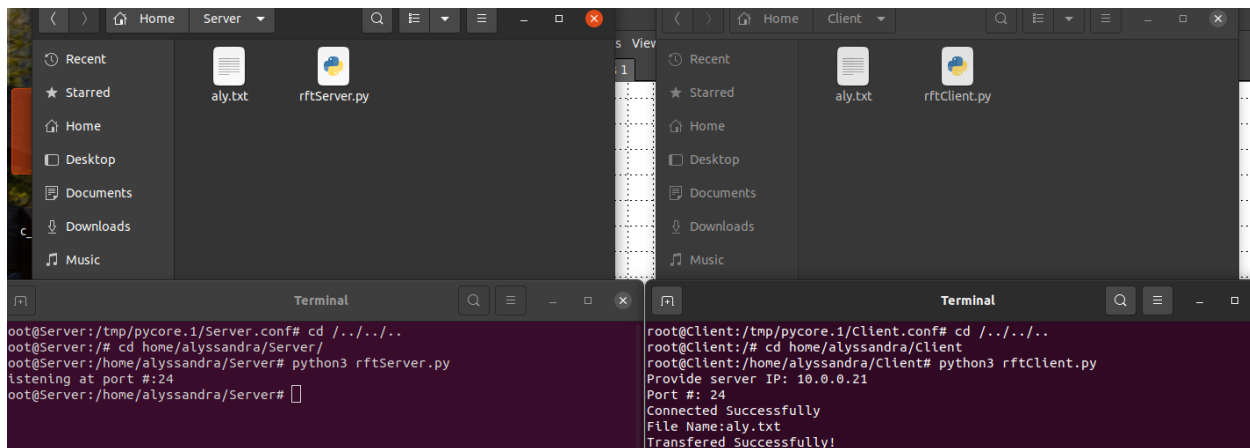


Figure 3. Figure 3 shows the output of the program after the .txt file was sent

- **Instructions for running the submitted programs:**

For running the programs, you must have the programs for client and a server on their respective file directory. After that, you need to set up the Client and the Server simulation on CORE. Finally, you open the terminals of the simulation and find the directory of their respective programs and run it there on the simulation.

- **References:**

[1] File Transfer using Socket Programming

<https://www.youtube.com/watch?v=SZyd7xGTBkw>

[2] Socket Programming in Python (Guide)

<https://realpython.com/python-sockets/>