Ayden Martin, Mitch Piazza, Wes Petrimoulx

CIS 457

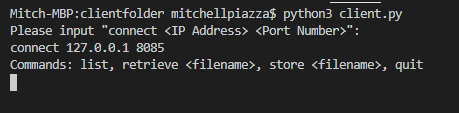
FTP server project

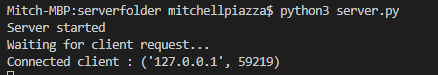
Our project is an FTP server written in python. The basic logic of the program is that you run server.py, then you run client.py. When you run client.py, it will ask you to input connect followed by an IP address and port number. When you connect, you will then be prompted with a print that tells you the commands: list, retrieve <filename>, store <filename> and quit. The list command will list all files in the directory of the server.py file. The retrieve command will take a parameter that is the file name, if the file exists, the file will then be brought to the same folder that client.py is in. the store command will take a parameter that is the file name, if the file exists, the file will be sent to the same folder that server.py is in. quit will cause the connection to terminate safely.

The main issue that we ran into was working with the retrieve function. We were able to establish connect and list easily, as well as quit. Retrieve was our biggest problem because we had to take time understand how to send and receive bytes using socket API. We were having trouble understanding how the bytes were sending, how to receive the bytes we did send, and using the buffers of socket. Once we were able to get an understanding of how those worked, we were able to get retrieve to work properly and send files from one directory to another. Since store is like retrieve, we did not run into many issues with the store functionality. Our two main problems were understanding socket API and writing retrieve function, which we solved by learning how the socket API worked.

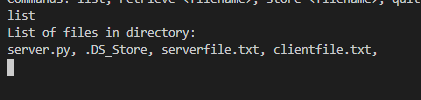
**Functions:**

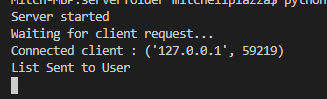
Connect:



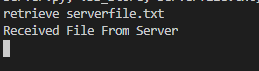


List:



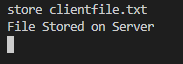


Retrieve:





Store:





Quit:

