Derek Teixeira Jigar Makwana Date: January 29, 2017

Code Explanation: classes and data types, pictures of operation

Server side code:

Start with (import socket) for using sockets in Python2.

Host will be set to default of 127.0.0.1

Port will be a random number of 5000

Most important line in using UDP will be line s = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM) DGRAM stands for datagram of a UDP

s.bind(host, port)) will bind the host and port together for the sockets

Infinite loop stays always true (s.recvfrom(1024)) is a buffer waiting for a response.

+ str(addr) will obtain and output the address of the person sending the data

+ str(data) will obtain and output the word sent form the client.

Str(data).upper() changes lowercase to uppercase.

s.sento(data, addr) will send back to the address sending the message.

s.close() will end the connection

Client side code:

Same host of 127.0.0.1

Port must be different, we go with 5001

We add the server which has a different port of 5000 to be recognized from the client.

s = socket and s.bind are the same as the Server.

We use s.sento(message, server) to get both the server and client to talk back in forth with one message sent

And end it again with an s.close()

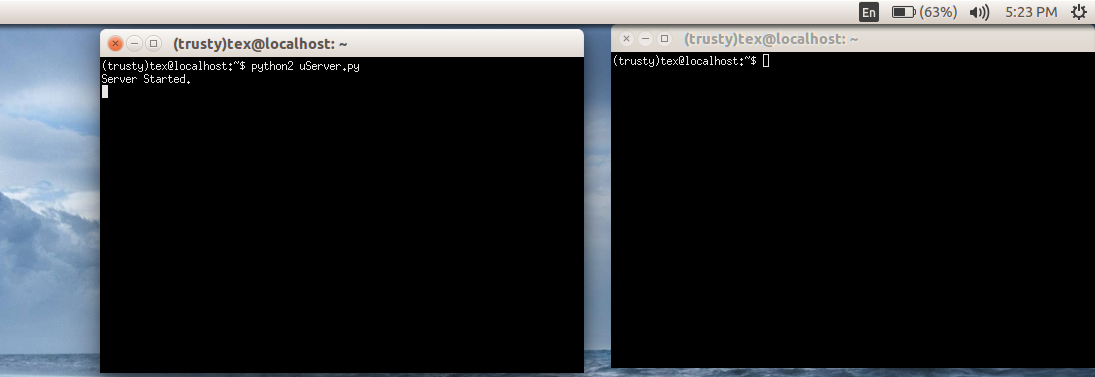
References Youtube video

https://www.youtube.com/watch?v=XiVVYfgDolU

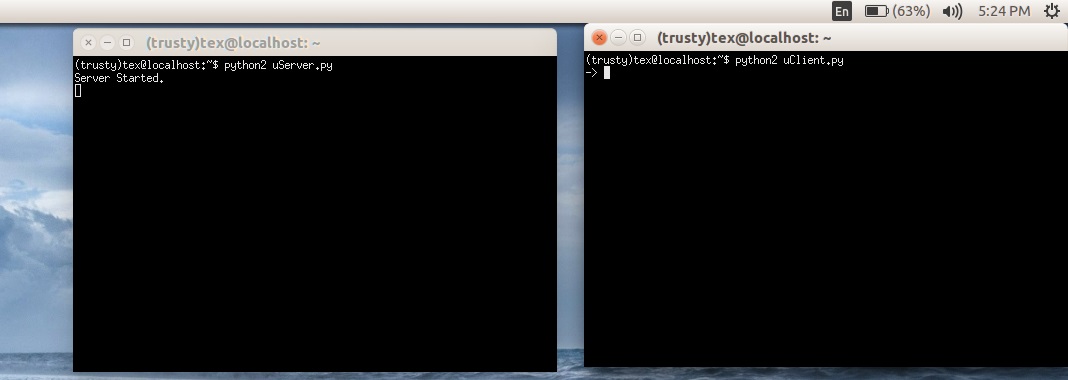
Pages 157-162 in Computer Networking A Top\_Down Approach by Kurose and Ross, Sixth Edition

Photos from Linux Ubuntu execution of Server and Client Files

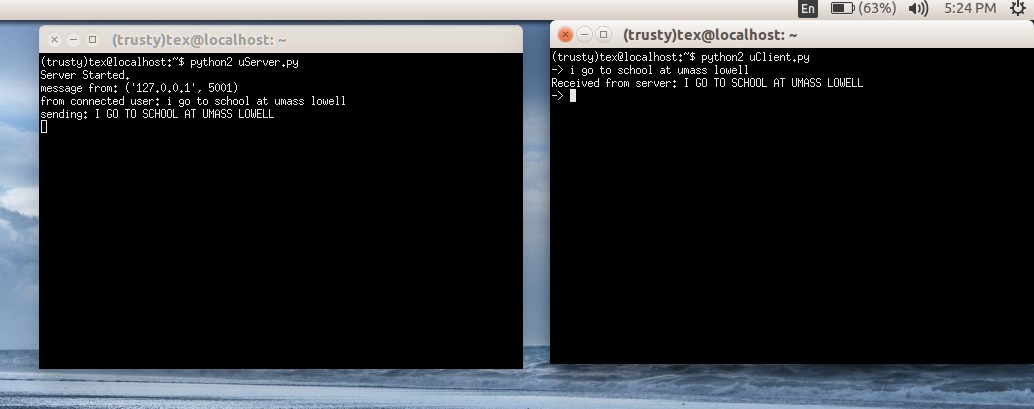
Step 1



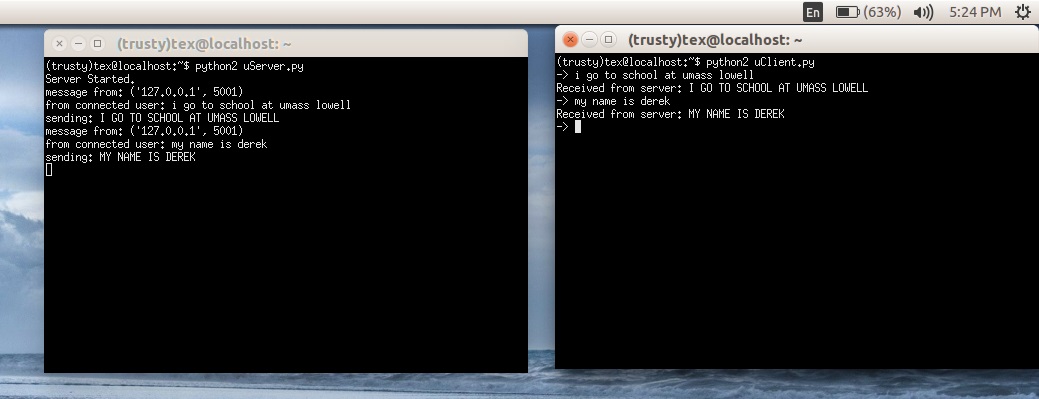
Step 2



Step 3

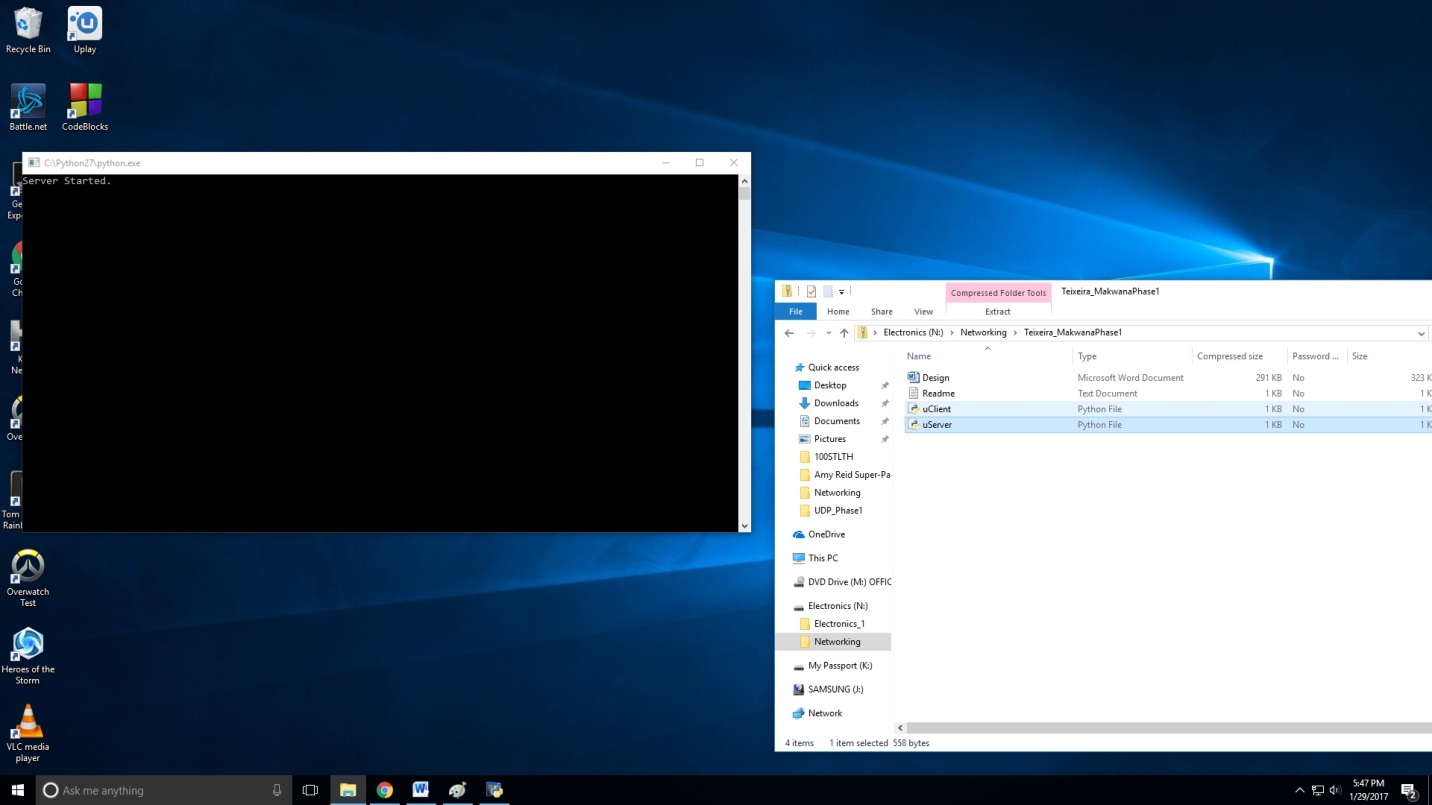


Step 4

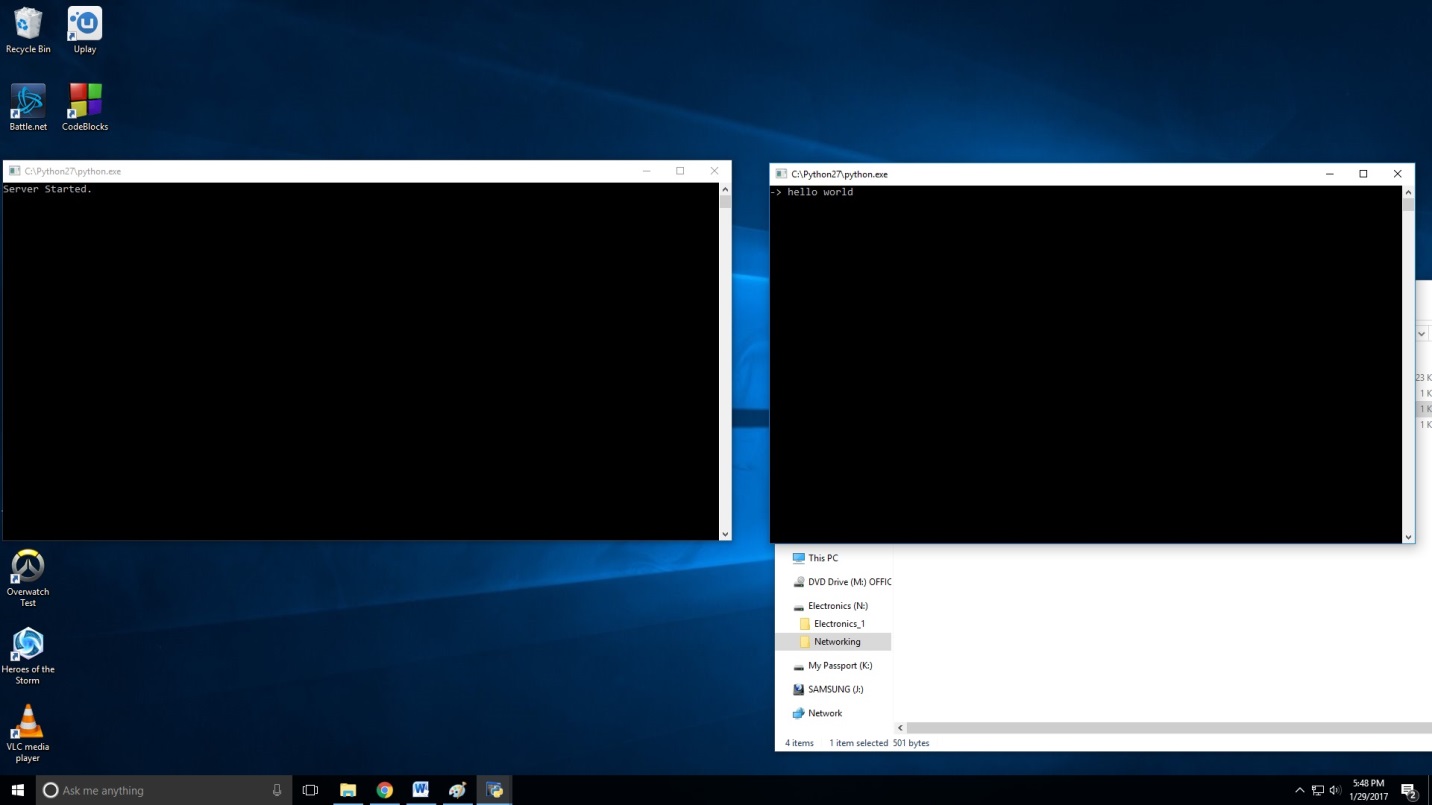


Execution in Windows 10

Step 1



Step 2



Step 3

