20163127612 여혁수 HW4 report

1) Describe your development environment information in detail (compilers/interpreter versions, compile options, used extra library etc.)

I implemented server and client program in Ubuntu 18.04.5 LTS version. Compilers/interpreter version is python 3.6.9 . You can set 5 client windows and 1 server window by input command ‘sudo ./execute\_mn.sh’ . Then, compile and run “server.py” by commanding ‘sudo python3 server.py’ . By this way, run “client.py” by commanding ‘sudo python3 client.py’ .

I import sleep module to implement timeout function of client. Also, I import literal\_eval module from ast library to transform string data to dictionary data.

2) Present how to design your assignment such as data structures and algorithms.

Server.py)

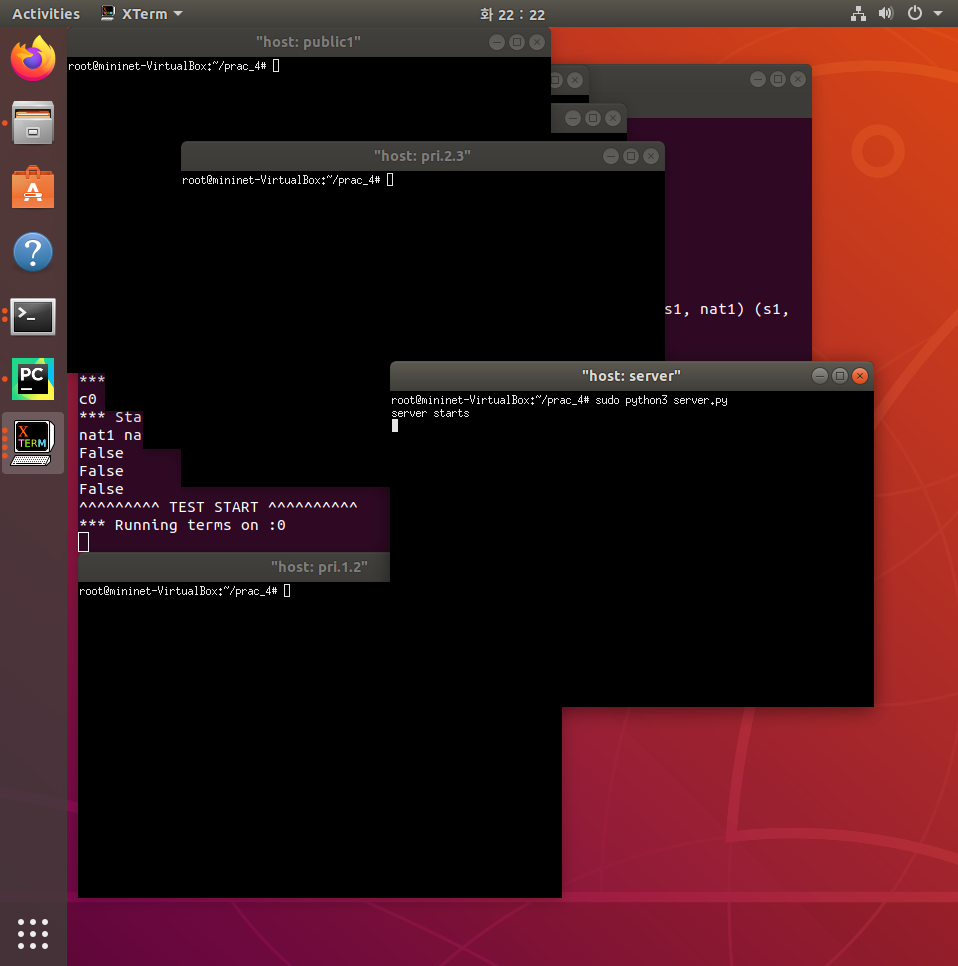
My server has 2 threads. One is to detect timeout of clients, another is to receive data. In thread of detecting timeout, continuously increase value in list of clients by 1 per one sec. Then, if some clients’ value go to 30, that clients are being timeout. So remove elements from list of clients. After removing, server sends new list to alive clients. In thread of receiving data, it can receive new client ID, request for ‘keep alive’, and deregistration request. Thread can differentiate what is goal of data by head part of data. I deliberately put header like “recvID”, “keep\_alive” to data. After differentiating goal, parse the data and properly send response to clients.

Client.py)

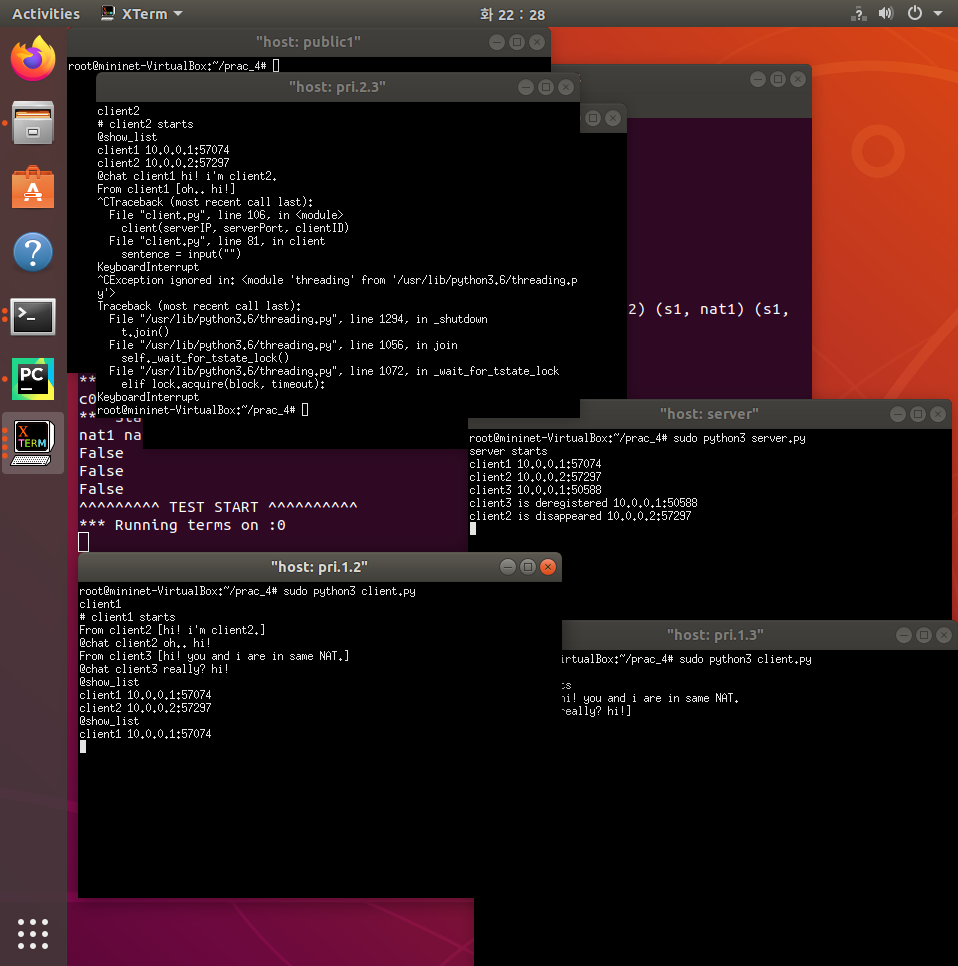
My client program has 2 threads, too. One is to receive data, and another is to send keep-alive-request to server. In thread of receiving data, client can receive updated list of clients from server, or any message from other clients. In thread of sending keep-alive-request, count 10 seconds and send keep\_alive request to server. Server gets this info and updates value which is increased by 1 per one sec, to zero. Except these, client program should have function of accepting command from terminal input. If “@show\_list” command came to client, just print information of alive clients from list. If “@chat” command came, parsing the command to get content of chat message and ID of receiver. Then check if the public NAT IP address of sender and receiver is same. If same, send message by using private IP address. If not, just send by using NAT IP.

I use dictionary data structure to store clients information. And I use list data structure to parse string data.

3) Explain how to run both sender and receiver programs including the screen capture.



If you input command “sudo ./execute\_mn.sh”, 6 windows will be generated. In window named “host: server”, enter the command like “sudo python3 server.py”. Server program will start.



You can run client program by entering the command in other windows like “sudo python3 client.py”. Other functionality is showed in this snapshot.