The information context about the chat program

Name：李政哲
ID：410285045
Grade：CSIE 2nd

1. System Environment

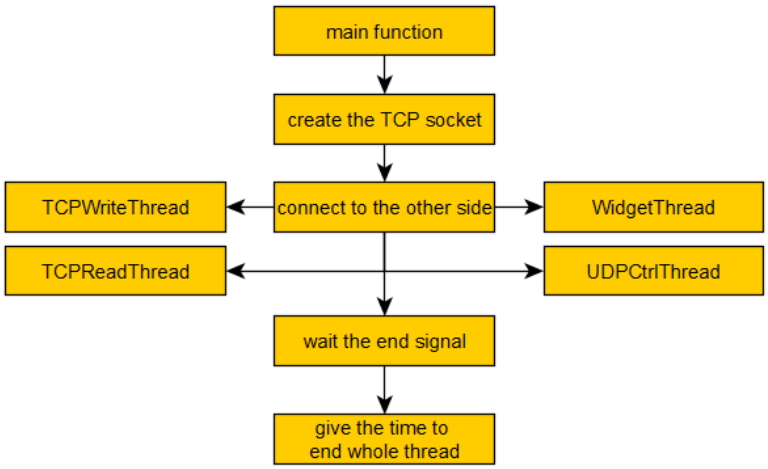
I use the Ubuntu 14.04 which is a branch of the Linux Gnome Debian system. In Window system, there is a common way that can call windows.h to create GUI. However, I didn't use windows. In my project, I use PyQt to get the same goal. As the different about my program, I use the OpenCV to manage the image too.

2. The Way To Compile

I use terminal in Ubuntu. The language I use to accomplish the program is Python.

i. Type " sudo python server1.py " on the server side
ii. Type " sudo python client1.py " on the server side

3. The Flowchart Of My Program

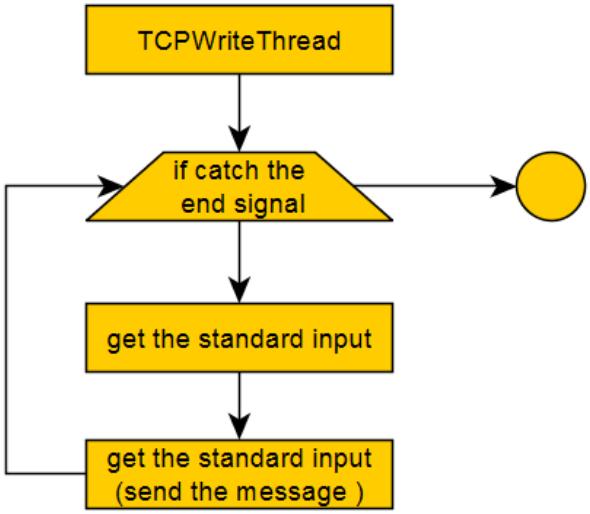


Flowchart1. The big structure of my program

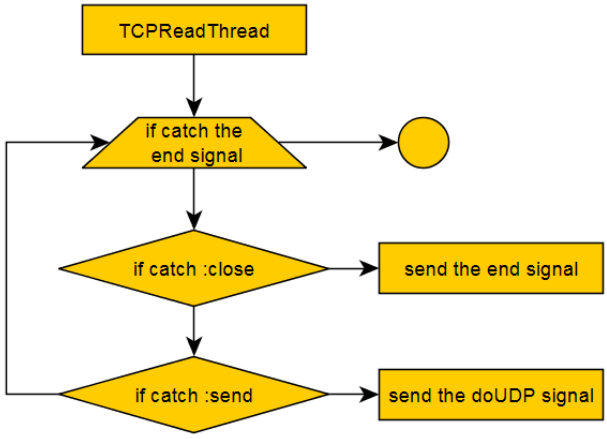
The above flowchart is the big structure of my program. As the problem request, I divide the program as four parts. I use two threads to deal with the TCP communication. The two threads would loop to detect if there are standard input of receiving the socket from the other side. The structure of the PyQt is OOP. So it would block the function. To solve this problem, I create a new thread

to do this.

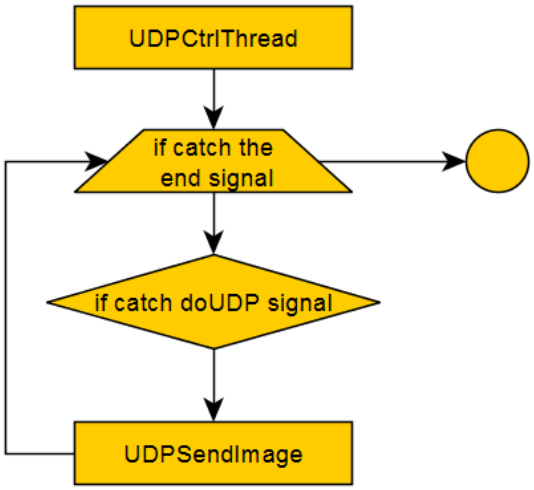
Why I should create the last thread? As my design, I would detected the key press event define in PyQt. But if we conduct the specific UDP function, the key press event function would be block, and GUI would be no response. So I create a new thread. The key press event function would send a change signal called “doUDP”. After sending it, the key press function would end itself. Second, the UDP thread would detect the “doUDP” signal. And it would toggle UDP sending function.



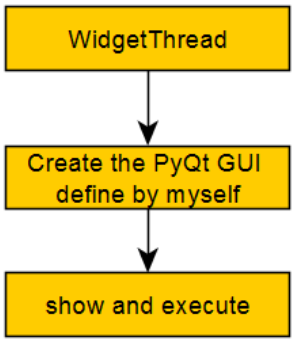
Flowchart2. The structure of TCP sending message thread



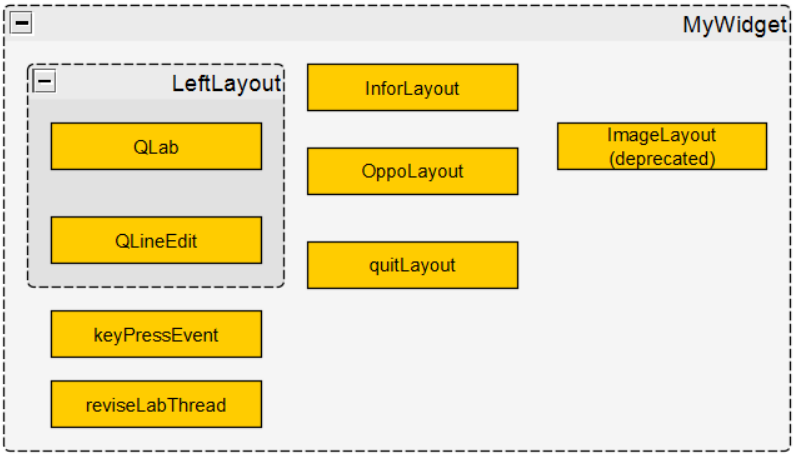
Flowchart3. The structure of TCP receiving message thread



Flowchart4. UDP thread structure



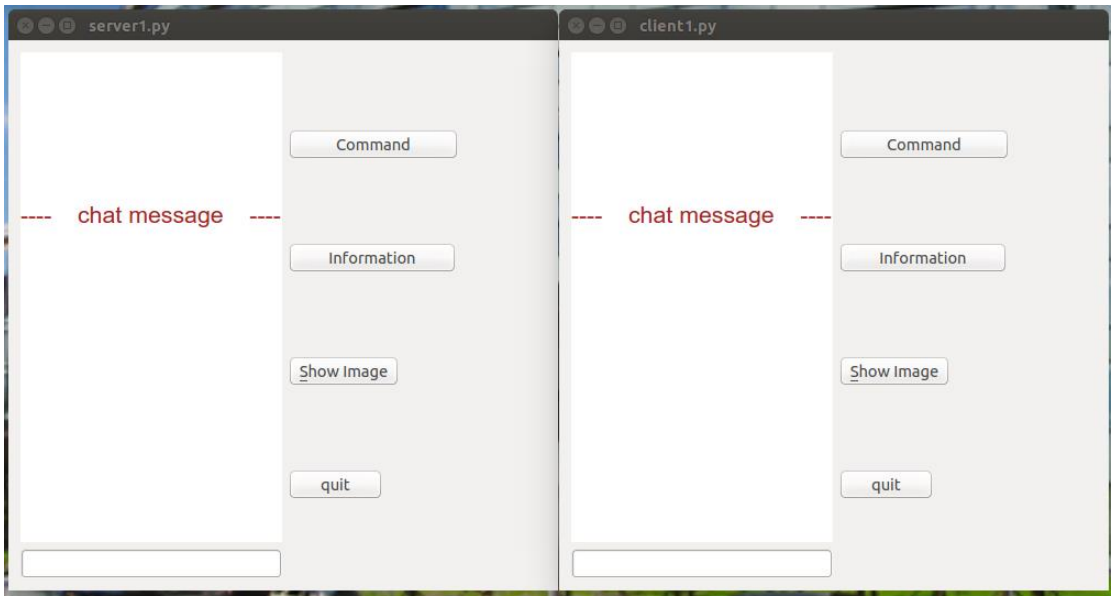
Flowchart5. GUI thread structure



Flowchart6. The structure of Layout class define by myself

As I mention, the principle of the PyQt is OOP. It's alike Java. So I define some method and the GUI widget. After the function "_exec()" conduct, it would detected if there're match event. The relative position show in flowchart is the around GUI that you can see below. The keyPressEvent is the event function that I define to interactive with the QEditLine. The reviseLabThread is the thread created in the constructor of the MyWidget object. It would detect the signal if it should change the text in the QLab object.

4. The Information To Use



Piture1. The client and server GUI

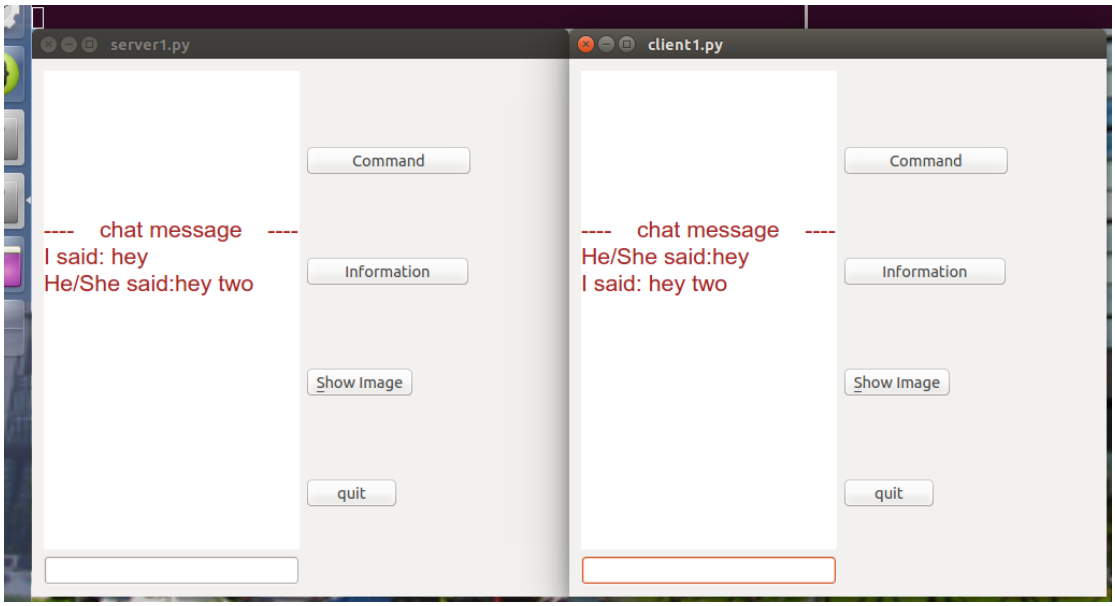
The picture shown above is the GUI made by PyQt. The left is server side; the other is the client side. The left part is the QLab that would show the whole conversation.

The right side has four buttons. First is Command button. When we press, it would show the command you can use. There are two command in my program: ":close" and ":send". When you type ":close", the connection would be close, and the program would end. Second, when you type ":send", it would enter the mode of sending image. The second button is information. When you press, it would show the other side's information. In this demo, I assign it as local host.

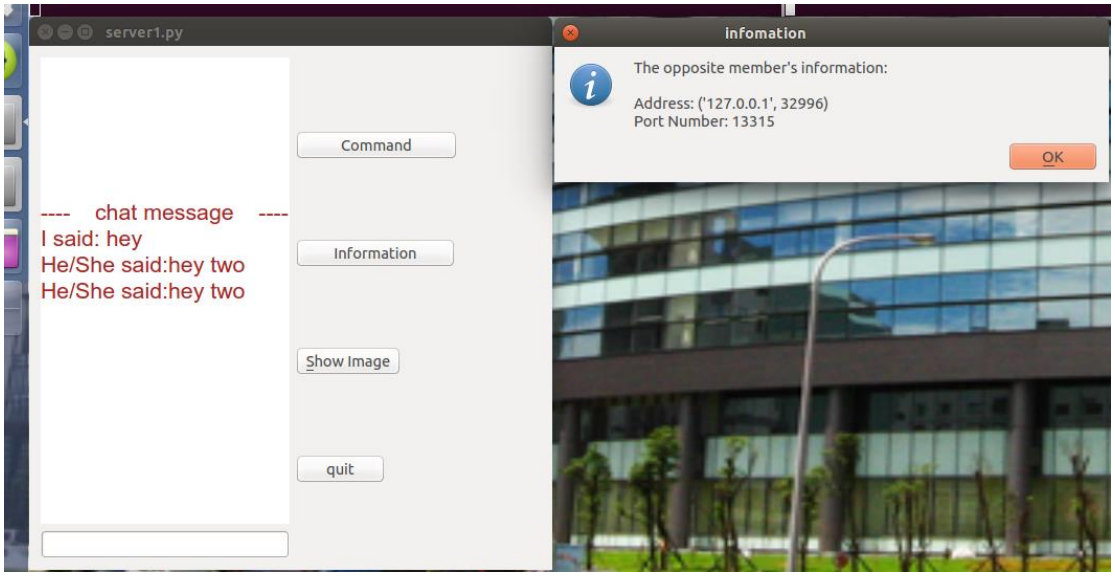
The third button is show image. After the other side sent you the image and you press it, it would show the image that had received. The last button is quit. The function of the quit button is the same as the command ":close".

How to use my GUI? First, you want to communicate with the others. So you can type the sentence in the edit line at the bottom left corner. After you finish, press the enter on your keyboard. The message would be sent to the other side, and shown on the other side's QLab. If you want to get the information that the button can give, you can press it any time. Notice! If the other side hasn't passed the image, you would get the warning if you press the third button.

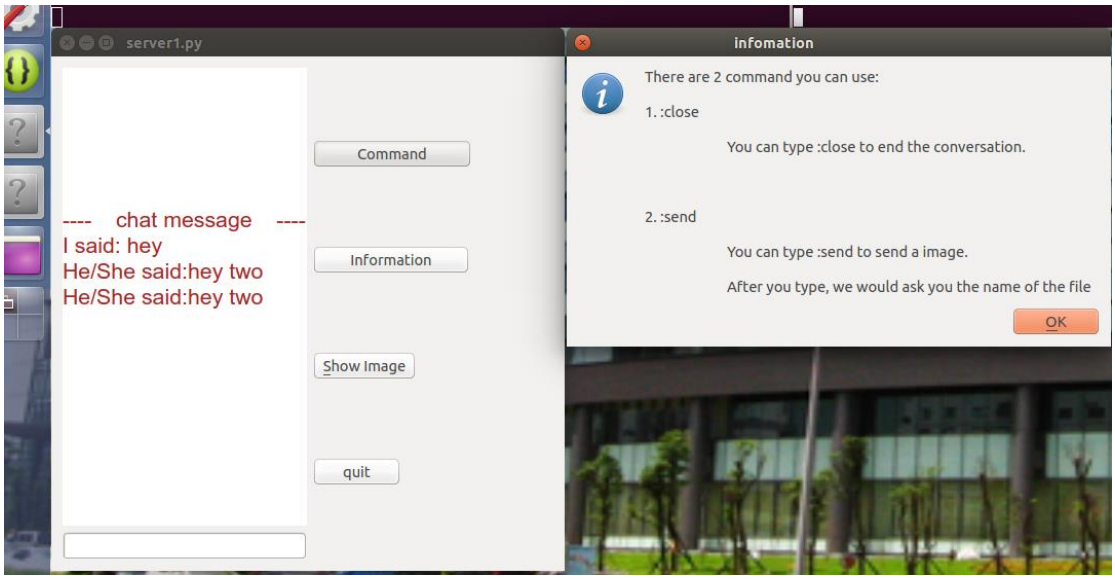
5. Result



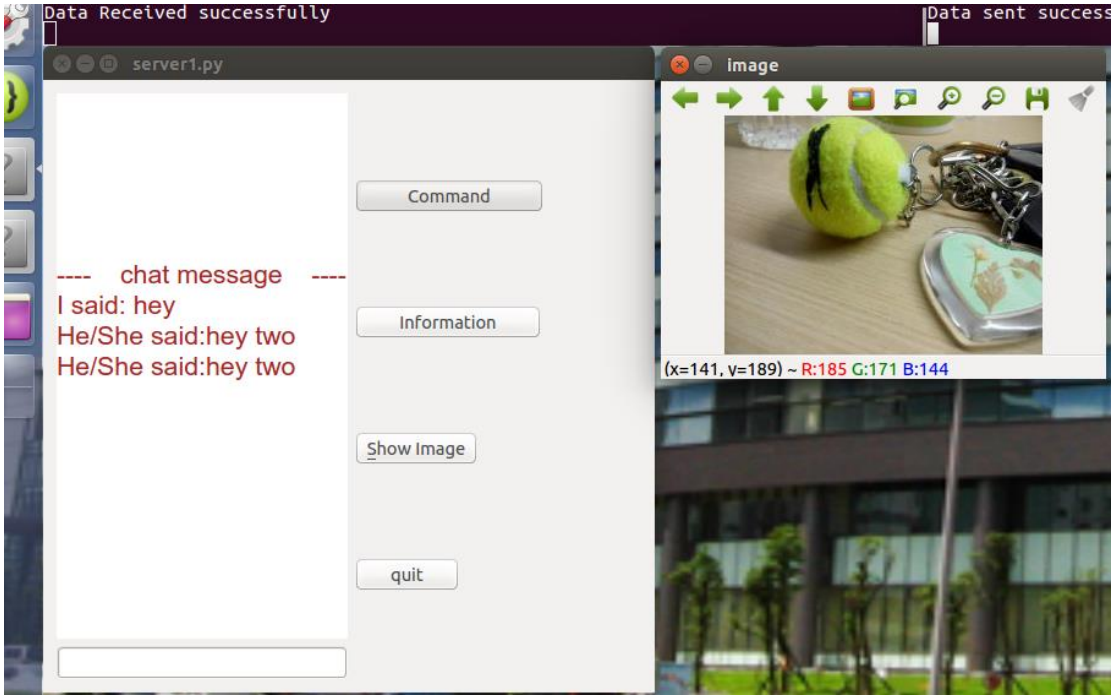
Picture2. The conversation process



Picture3. Press the information button



Picture4. Press the command button



Picture5. Show the image after transferring