[106-2] Computer Network – HW2

TA 黃學亞 r05921057@ntu.edu.tw

Due: 4.15 (Sun.) 21:00

1. 本次作業請於期限內至 ceiba 作業區上傳，格式為單一 zip 檔，解壓縮後應恰為一個以學號為名的資料夾，資料夾內有一以學號為檔名的 pdf 檔及四子資料夾，如下例所示。
   ---inside *.zip
   r05921057.pdf
   r05921057/src_P1/socket_client.py
   r05921057/src_P2/*
   r05921057/src_P3_a/*
   r05921057/src_P3_b/*
2. 逾期繳交不予計分，請務必盡早開始，並以自己的努力完成。
3. Problem 1 的 client 最後需與助教於工作站架設的 server 連線並傳送你的正確資料(注意步驟 V 之後的操作)，若未完成此步驟將斟酌扣除部份成績。
4. 如有任何相關問題歡迎來信，並預祝作業練習順利！

Problems:

1. [Socket Programming - TCP]

With a TCP Server provided to you, we're going to construct a client to communicate with it. Fig. 1-1 describes the messages sent between client and server.

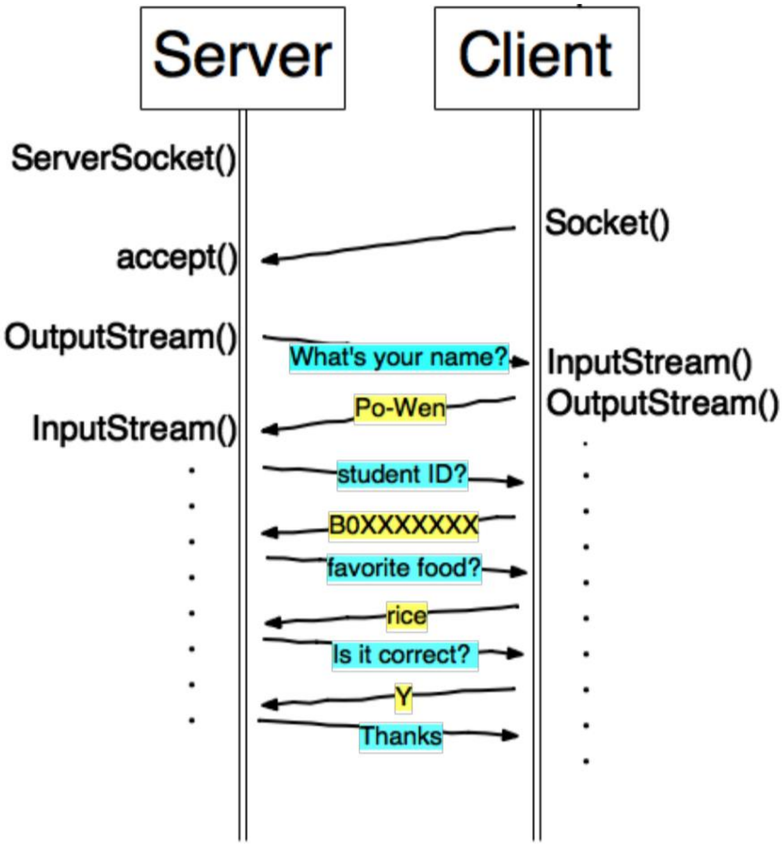


Fig. 1-1 Message flow between server and client

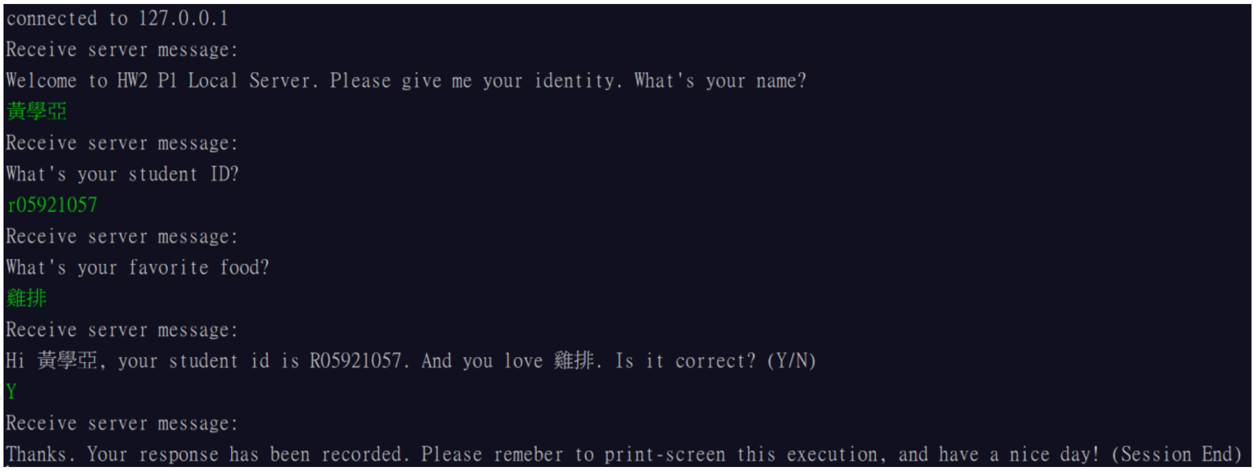
Please refer to the following steps:

I. Read the source code of socket_server.py (works on python2 or 3), try your best to understand it.

II. Write your own socket_client.* such that it can 1) read the message sent from server and 2) send the user-input message to the server. There will be 5 messages from server and 4 messages sent to the server alternatively like Fig 1-1. socket_client.* can be written in any code as long as you can communicate with our socket_server which was written in python. C++ and python as client has both been tested.

III. Compile and run the socket_server.py first, and then run your socket_client.*.

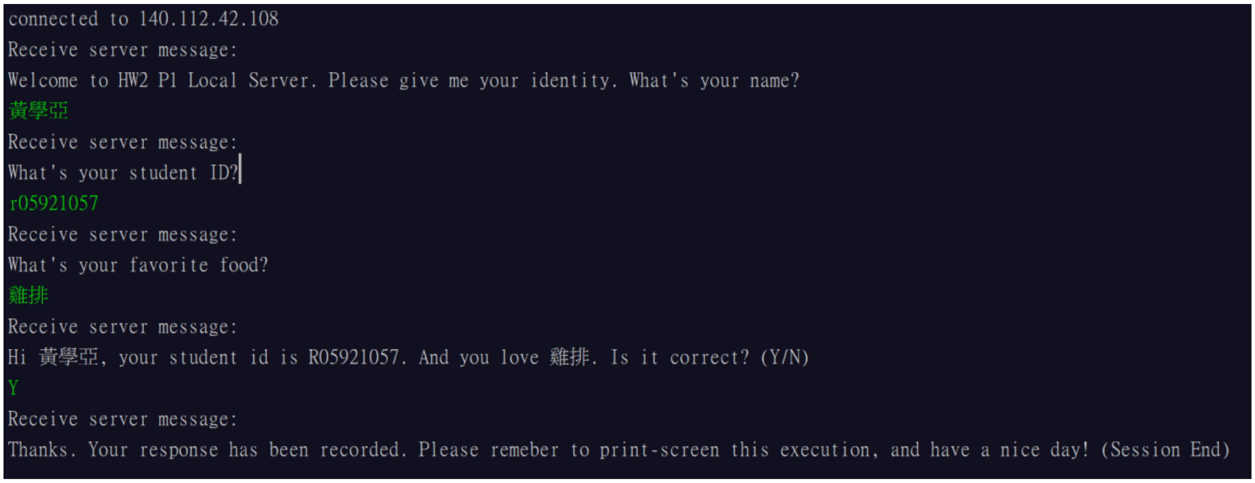
IV. Test it under your local machine. If you do every correct, the execution should be like Fig. 1-2. Make sure your program runs successfully before jumping to the next step.



```
connected to 127.0.0.1
Receive server message:
Welcome to HW2 P1 Local Server. Please give me your identity. What's your name?
黃學亞
Receive server message:
What's your student ID?
r05921057
Receive server message:
What's your favorite food?
雞排
Receive server message:
Hi 黃學亞, your student id is R05921057. And you love 雞排. Is it correct? (Y/N)
Y
Receive server message:
Thanks. Your response has been recorded. Please remeber to print-screen this execution, and have a nice day! (Session End)
```

Fig. 1-2 Connect to server on local machine(127.0.0.1)

V. In socket_client.*, try to connect to a socket on “140.112.42.108” with port 12345. Execute it, and input your real identity information. After you get similar results like Fig. 1-3, please print---screen it and attach it into your submitted *.pdf.



```
connected to 140.112.42.108
Receive server message:
Welcome to HW2 P1 Local Server. Please give me your identity. What's your name?
黃學亞
Receive server message:
What's your student ID?
r05921057
Receive server message:
What's your favorite food?
雞排
Receive server message:
Hi 黃學亞, your student id is R05921057. And you love 雞排. Is it correct? (Y/N)
Y
Receive server message:
Thanks. Your response has been recorded. Please remeber to print-screen this execution, and have a nice day! (Session End)
```

Fig. 1-3 Connect to our online server(140.112.42.108)

VI. Please put your source code under [student_id]/src_P1/socket_client.* Congratulation!

2. [Socket Programming - UDP]

a. Consider a UDP server and a UDP client. If the client sends 10000 packets back to back to the server (i.e. continually send 10000 packets), how many packets will the server receive? And how many of them are lost? Please design an experiment running on your local machine.

b. **(Bonus)** Make your client and server run on different machine as far as you can (through wireless transmission will be better). Compare with the result with both client and server running on your local machine.

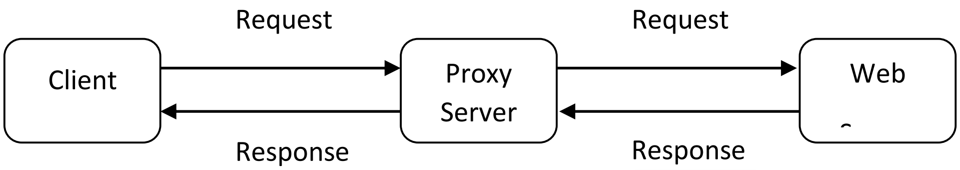
Please answer the above questions based on your own result on the *.pdf, and comment on it. Please also print-screen your execution and attach it to *.pdf. Finally, put your source code under [student_id]/src_P2/.

3. [Web Server]

a. In this assignment, you will develop a simple Web server in Python that is capable of processing only one request. Specifically, you Web server will (i) create a connection socket when contacted by a client (browser); (ii) receive the HTTP request from this connection; (iii) parse the request to determine the specific file being requested; (iv) get the requested file from the server's file system; (v) create an HTTP response message consisting of the requested file preceded by header lines; and (vi) send the response over the TCP connection to the requesting browser. If a browser requests a file that is not present in your server, your server should return a "404 Not Found" error message.

You can start from rewriting the socket_server.py to be run the server on your tested machine with port 80. Put a html file under the directory you run your server, and get the content through your browser. Please print-screen the result and attach it to *.pdf. Finally, put your source code under [student_id]/src_P3_a/.

b. **(Bonus)** In this assignment, you will develop a Web proxy. When your proxy receives an HTTP request for an object from a browser, it generates a new HTTP request for the same object and sends it to the origin server. When the proxy receives the corresponding HTTP response with the object from the origin server, it creates a new HTTP response including the object, and sends it to the client. This proxy will be multi-threaded, so that it will be able to handle multiple requests at the same time.



The skeleton code is provided to you in the attached file with file name proxy_server.py. Your job is to complete the code, and then test it by having different browsers request Web objects via your proxy. Please print-screen the result and attach it to *.pdf. Finally, put your source code under [student_id]/src_P3_b/.