**ใบงานที่ 5 Thread**

1. ใบงานนี้จะเป็นการศึกษาการพัฒนาโปรแกรมแบบมัลติเทรด คือการทำให้โปรแกรมแม่ข่ายสามารถรองรับการเชื่อมต่อจากลูกข่ายได้มากกว่า 1 connection โดยให้นักศึกษาสร้างไฟล์ชื่อ ThreadServerxxx.py(xxx=รหัสนักศึกษา 3 ตัวหลัง)

คัดลอกโค๊ดต่อไปนี้ไปใส่

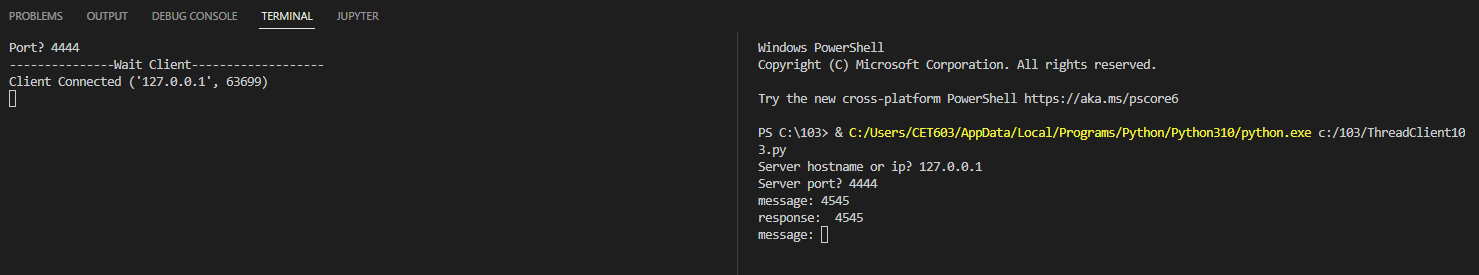


**ใบงานที่ 5 Thread**

1. สร้างไฟล์ชื่อ ThreadClientxxx.py(xxx=รหัสนักศึกษา 3 ตัวหลัง)



1. รันโปรแกรม ThreadServer ระบุพอตที่ต้องการจะเปิดรับการเชื่อมต่อ(1024-65535) จากนั้นรับโปรแกรม ThreadClient ระบุ IP และ Port ที่จะเชื่อมต่อไปหา ทดลองพิมพ์ข้อความเพื่อส่งไปหา ThreadServer บันทึกผลการรันของทั้ง 2 โปรแกรม

**บันทึกผลการรัน........................................**

1. แก้ไขให้ข้อความที่รับกลับมาจาก ThreadServer จากเดิมมี b’ ติดมาด้วยให้เหลือเพียงข้อความปกติ

**บันทึกโค๊ดที่แก้ไขแล้ว...............................import socket**

**sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)**

**host = input("Server hostname or ip? ")**

**port = int(input("Server port? "))**

**sock.connect((host, port))**

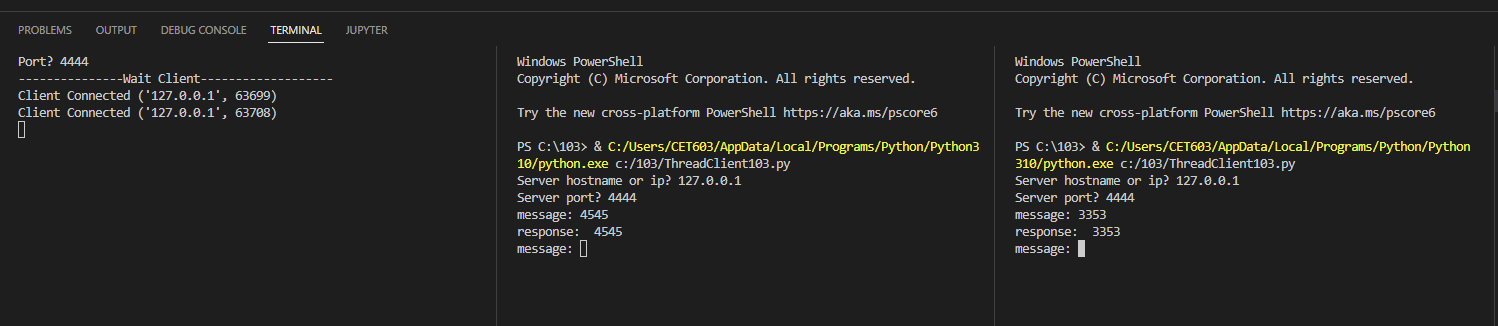
**while True:**

**data = input("message: ")**

**sock.send((data).encode("utf-8"))**

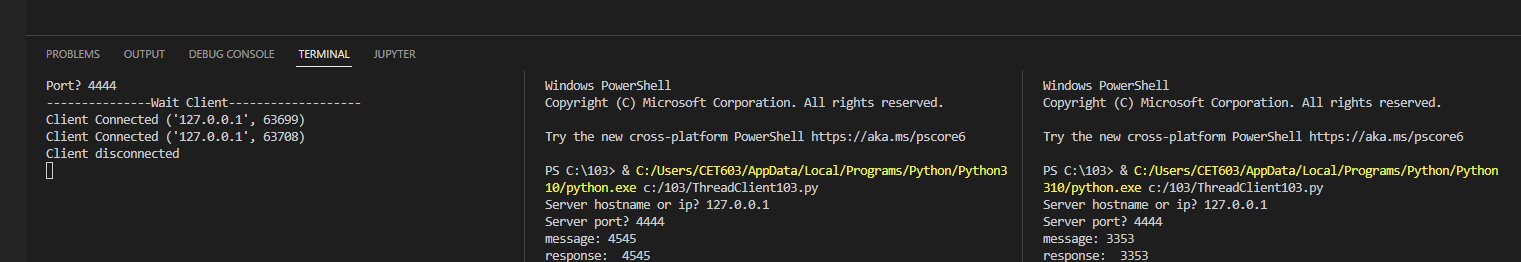
**print("response: ", sock.recv(1024).decode("utf-8"))**

1. ทดลองรัน ThreadClient เพิ่มเป็น 3 Connect ทดลองส่งข้อความจากทุก Client ที่รันอยู่

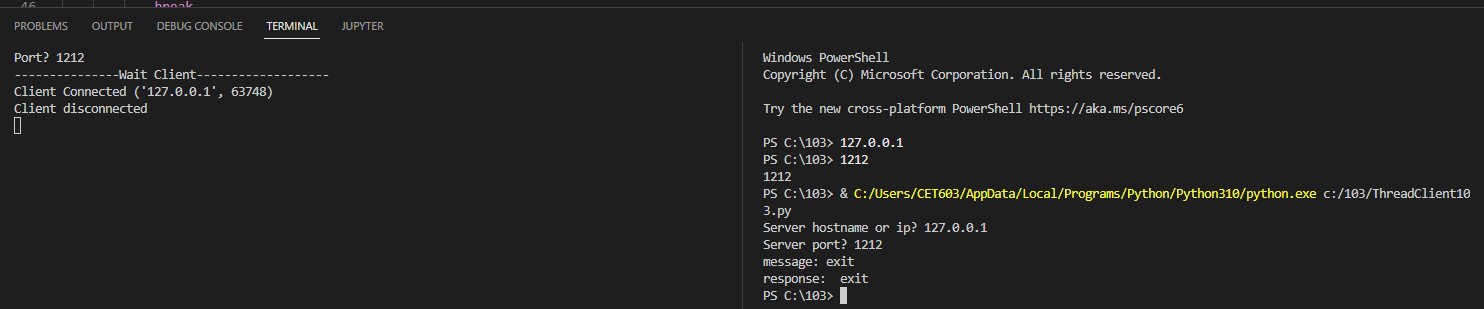
**บันทึกผลการรัน........................................**

1. ทดลองออกจากโปรแกรม ThreadClient ทุกตัวโดยการกด Hotkey = Ctrl+c

**บันทึกผล ThreadClient และ ThreadServer................**

****

1. จากข้อ 6 ให้แก้ไขโปรแกรม ThreadClient โดยเมื่อผู้ใช้งานพิมพ์ @exit ให้ออกจากโปรแกรมสั่งเกตุผลที่เกิดกับ ThreadServer บันทึกผลและอธิบายว่าเหตุใดจึงเป็นเช่นนั้น

**บันทึกผล...............เพราะเป็นการหยุดการเชื่อมต่อ เลยเกิดการ disconnected**

1. ให้นักศึกษาแก้ไขให้ ThreadServer มีการแสดง IP และ Port **ของ** Client ที่เชื่อมต่อเข้ามา และให้ ThreadServer ส่งข้อความต้อนรับไปยัง ThreadClient เพื่อบอกให้ ThreadClient ทราบว่า ThreadClient เชื่อมต่อเข้ามาเป็นลำดับที่เท่าไหร่ตั้ง ThreadServer เริ่มทำงาน

**บันทึกโค๊ด และ ผลการทำงาน........................**

**โค้ด ฝั่ง** ThreadServer

from distutils.log import error

from http import client

import socket

import threading

import os

class ThreadedServer(object):

    def \_\_init\_\_(self, host, port,):

        self.host = host

        self.port = port

        self.sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

        self.sock.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1)

        self.sock.bind((self.host, self.port))

    def listen(self):

        print("---------------Wait Client-------------------")

        self.sock.listen(1)

        data1 = "อันดับการติดต่อ"

        i = 1

        while True:

            client, address ,  = self.sock.accept()

            data = ''

            print("Client Connected",address)

            client.settimeout(60)

            threading.Thread(target=self.listenToClient,

                             args=(client, address)).start()

            print(data1,+ i )

            client.send(bytes(data1,"utf-8"))

            client.send(bytes(str(i),"utf-8"))

            i += 1

    def listenToClient(self, client, address):

        size = 1024

        while True:

            try:

                data = client.recv(size)

                if data:

                    # Set the response to echo back the recieved data

                    response = data

                    client.send(response)

                else:

                    raise error('Client disconnected')

            except:

                client.close()

                return False

if \_\_name\_\_ == "\_\_main\_\_":

    os.system("cls")

    while True:

        port\_num = input("Port? ")

        try:

            port\_num = int(port\_num)

            break

        except ValueError:

            pass

    ThreadedServer('', port\_num).listen()

**โค้ด** Threadclient

from re import S

import socket

import os

sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

host = input("Server hostname or ip? ")

port = int(input("Server port? "))

sock.connect((host, port))

d1 = ''

d2 = sock.recv(1024)

d3 = ''

d4 = sock.recv(1024)

d1 += d2.decode("utf-8")

d3 += d4.decode()

print(d1+ "  : " + d3)

while True:

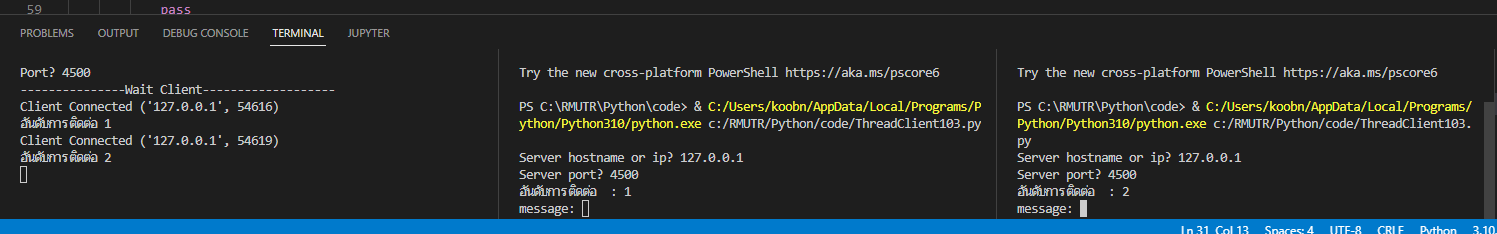
    data = input("message: ")

    sock.send((data).encode("utf-8"))

    print("response: ", sock.recv(1024).decode("utf-8"))

sock.close()

**ผลการทำงาน**

****

1. ให้แก้ไข ThreadServer ให้สามารถแสดงข้อความที่ ThreadClient ส่งเข้ามาโดยให้ระบุว่าเป็นข้อความของ ThreadClient ใดส่งมาจากชื่อที่ผู้ใช้กรอกที่ ThreadClient นอกจากนี้ที่โปรแกรม ThreadClient จะต้องสามารถรับข้อความจาก ThreadClient อื่นๆ ที่เชื่อมต่อเข้ามาได้เช่นเดียวกันที่ ThreadServer ได้รับ ดังนั้น ThreadClient ที่เชื่อมต่อเข้ามาทุกคนจะเหก็นข้อความของ ThreadClient ที่ส่งเข้าไปทุกคน เช่นเดียวกัน ThreadServer(โปรแกรมแชท) \*\*\*ให้นักศึกษาปรับปรุงจากโปรแกรมเดิม ห้ามไปคัดลอกโปรแกรมใหม่จากที่อื่นมาเปลี่ยน แต่นำมาใช้งานร่วมได้