

Report Socket Programming

Socket api เป็นapiไว้สำหรับติดต่อสื่อสารระหว่าง clientหรือ serverโดยใช้protocol TCPหรือ protocol UDPการใช้ทั้งสอง protocolจะเขียนไม่เหมือนกัน และTCP socket จะมีคุณลักษณะที่เพิ่มมาจากUDP คือ 1.การตรวจจับว่ามีpackageใดตกหล่นไหม แล้วเมื่อเจอก็จะส่งซ้ำไป 2.เมื่อส่งซ้ำไปจากการหล่นหายแล้ว TCPจะจัดเรียงข้อมูลใหม่ได้เองโดยอัตโนมัติ ซึ่งต่างจากUDPถ้าหากจะส่งซ้ำจะต้องเขียนโปรแกรมเอง

การส่งด้วยTCPนั้นจะมี socket สองฝั่งคือsocketที่ฝั่งserverสร้างขึ้นมาเพื่อbindและlisten และจะมีอีกsocketคือฝั่งclientส่งมาและรอฝั่งserver accept จากนั้นclient ส่งdataและฝั่งserverก็จะรับและส่งกลับเป็นข้อความเดิมที่ส่งมาและก็ปิดsocket server นั้นลง แต่โค้ดข้างต้นในเว็บนั้นsocket serverจะถูกปิดลงทุกครั้งหลังจากที่client ทำงานเสร็จแล้ว หากจะแก้ไขให้สามารถทำงานได้หลายๆครั้งต้องเพิ่มwhile True ก่อนที่server socketจะaccept ให้clientเชื่อมต่อกับserverได้

```
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    while True:
        conn, addr = s.accept()
        with conn:
            print(f"Connected by {addr}")
            while True:
                data = conn.recv(1024)
                if not data:
                    break
                conn.sendall(data)
```

ดั่งโค้ดข้างต้นนี้จะเป็นการแก้ไขให้ทำงานได้ต่อ เมื่อsocketแรกที่ย้ายไปทำงานเสร็จแล้วสามารถทำได้ไปเรื่อย ๆ จนกว่าจะหยุดการทำงาน มีwhile loop Trueข้างในอีกชั้นคือ การรับข้อมูลและตรวจสอบว่าข้อมูลที่ได้นั้นมีไหมถ้าหมดแล้วก็จะออกจากลูปไปทำงานrequestต่อไปของclient แต่ถ้ามีข้อมูลส่งมาก็จะส่งข้อมูลนั้นกลับไปให้client

ทั้งหมดข้างต้นเป็นการเชื่อมต่อแบบ 1-1 คือสามารถเชื่อมต่อได้แค่clientเดียว

Multiple-Connection Client คือการที่server สามารถเชื่อมต่อได้กับclientหลายๆตัวในเวลาเดียวกัน ซึ่งข้างต้นที่แก้ไขข้างบนนั้นเป็นการแก้ไขให้สามารถรับได้หลายclientแต่ไม่ใช่เวลาเดียวกัน หลักการทำmultiple-Connectionคือ การsetblockingเป็น falseเพื่อไม่ให้socketเป็นblocking function อีกต่อไป แล้วและไปใช้การselectorแทนในการacceptแทน และทำงานโดยใช้while True กำหนดevents = selector และfor key,mask in events เมื่อมีeventsเข้ามา while loopก็จะทำงานโดยจะเริ่มทำที่accept_wrapperก่อนเพราะถ้าเข้ามาครั้งแรกจะยังไม่มีdataต้องaccept เมื่อacceptจะต้องกำหนดsetblockingเป็น falseอีกครั้งเพราะเป็นคนละsocket socketที่เข้ามานี้เป็นของclient และ เพิ่มdataเข้าไปทั้งReadและwrite เมื่อวนลูปใหม่ก็จะเป็นการเชื่อมต่อเพราะมีdataอยู่ข้างในหรือผ่านกระบวนการacceptมาเรียบร้อยแล้ว function ต่อไปจะเป็นการอ่านและส่งข้อมูล