TCP-programmering med Python del 2

Formål: At lave mere avancerede TCP-programmering i Python

Baggrund:

- TCP opgave 1

Opgave 1.1: Simpel Echo Server – i Python

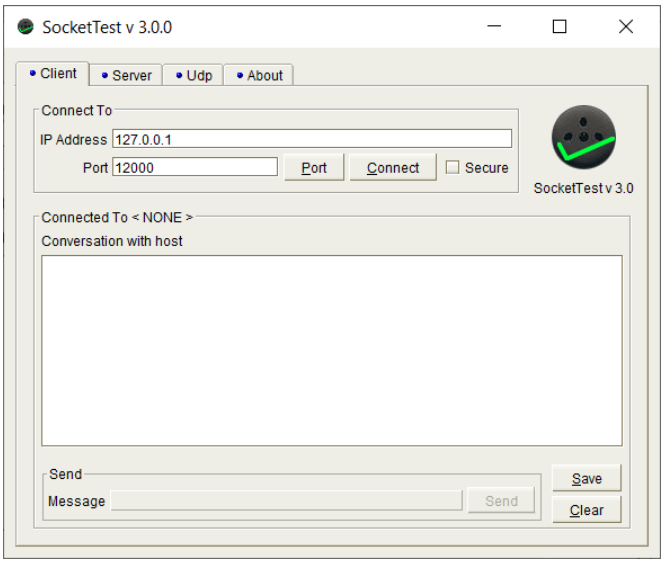
Du skal lave en simpel ekko-server (se RFC862 for yderligere information) – denne gang i Python

I bogen "Computer Networking" kap 2.7.2: ss.193-194 er et simpelt ekko program beskrevet.

Lav et nyt python-projekt og slet det der står i main.py. Tag og indskriv (kopier) det der står i bogen s 193-194.

Opgave 1.2: Test din Echo Server

Du skal nu benytte SocketTest til at test din Simpel Echo Server. Start SocketTest – og du får:



Indsæt postnummeret og 'conenct', du kan nu skrive en 'message' og klik 'send' for at sende den til din server og få et svar.

Opgave 2.1: Refaktor Serveren til at håndtere flere klienter samtidigt

Din server kan behandle en klient ad gangen og først derefter behandle den næste.
Du skal nu lave din server så den kan håndtere flere klienter samtidigt.

Step 1:
Flyt al kode, der har en én klient at gøre, over i en metode (kald den evt. handleClient).
den skal have to parametre en socketConnection og addr (de to variable der bliver initialiseret ved 'accept'-kaldet).
Hint: det er al kode efter accept-kaldet der flyttes til denne metode.

Din kode skulle gerne lige noget a la dette:

```
while True:
    connectionSocket, addr = serverSocket.accept()
    handleClient(connectionSocket, addr)
```

og din metode noget a la dette:

```
def handleClient(connectionSocket, address):
    sentence = connectionSocket.recv(1024).decode()
    print(sentence)
    capitalizedSentence = sentence.upper()
    connectionSocket.send(capitalizedSentence.encode())
    connectionSocket.close()
```

Prøv om dit program stadig virker.
Start din server og start to udgaver af SocketTest og send en 'message' fra den senest åbne SocketTest – hvad sker der?
Så send en fra den først åbne – hvad sker der nu?

Step 2:

Du skal nu I din server understøtte at der kan køre flere udgaver samtidigt (altså benyt Threading)

I Python gøres dette ved:

1. import threading # gøre brug af threading modulet
2. Du starter en thread ved:

   ```
   threading.Thread(target = <function>,args = (<args>)).start()
   ```

   Udskift <function> med din metode og <args> med *connectionSocket, addr*
3. Implementer dette i din server

Prøv den same øvelse som før med at åbne to SocketTest og se hvad der skar nu.