

### Exercice 1. Serveur TCP

Nous avons vu la manipulation de socket UDP, avec l'envoi et la réception de message. Nous allons maintenant utiliser les sockets TCP, qui utilise un flux de données.

Le serveur suivant contient un compteur dont on peut obtenir la valeur en lui envoyant la requête `get`.

**1.1** Recopier ce code, et consulter la documentation python pour comprendre les différentes instructions, en particulier la création de sockets en TCP, la méthode `makefile`, et les méthodes de lecture et d'écriture.

<https://docs.python.org/3/library/socket.html>

<https://docs.python.org/3/glossary.html#term-file-object>

```
#!/usr/bin/python3

import socket

class Server :

    def __init__(self):
        self.counter=0

    def mainServer(self, port):
        sock = socket.socket()
        sock.bind(("0.0.0.0", port))
        sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
        sock.listen(10)
        while True:
            cli, addr = sock.accept()
            sess = Session(self, cli)
            sess.mainSession()

class Session:

    def __init__(self, server, sock):
        self.server = server
        self.socket = sock
        self.file = sock.makefile(mode="rw")

    def mainSession(self):
        while True:
            line = self.file.readline().strip()
            if line == "get":
```

```
        self.file.write("val %d\n" % self.server.counter )
        self.file.flush()
    elif line == "quit":
        self.file.write("quit\n")
        self.file.flush()
        break
    else:
        self.file.write("err\n")
        self.file.flush()

    self.file.close()
    self.socket.shutdown(socket.SHUT_RDWR)
    self.socket.close()
```

```
Server().mainServer(5555)
```

Tester votre serveur avec netcat

```
netcat localhost 5555
```

## Exercice 2. Client

Voici un exemple de code client.

```
#!/usr/bin/python3
import socket

def client (host , port ):
    sock = socket.socket()
    sock.connect((host , port))
    f = sock.makefile(mode="rw")
    f.write("get\n")
    f.flush()
    print (f.readline() , end="")
    f.write("quit\n")
    f.flush()
    print(f.readline() , end="")
    f.close()
    sock.shutdown(socket.SHUT_RDWR )
    sock.close()

client ("localhost" , 5555)
```

**2.1** Recopiez le code et tester votre serveur.

**2.2** Modifier le code du client pour qu'il lise sur l'entrée standard la commande du client, et l'envoi eu serveur, tant que l'utilisateur n'entre pas quit. Pensez à gérer les instructions non prévues par le serveur.

### Exercice 3. Extension du serveur

#### 3.1 Ajoutez les opérations suivantes sur le serveur

- `incr` incrémente le compteur et retourne la nouvelle valeur
- `decr` décrémente le compteur et retourne la nouvelle valeur
- `add N` ajoute N au compteur et retourne la nouvelle valeur