TP4 : Programmation Réseau Année : 2022/2023

1. Compléter les deux bouts de programmes ci-dessous.
2. Tester ces deux programmes
3. Modifier le programme qui convient en utilisant l’extension de la classe thread.

import socket

IP = socket.gethostbyname(socket.gethostname())

PORT = 5656

ADDR = (IP, PORT)

FORMAT = "utf-8"

SIZE = 1024

DISCONNECT\_MSG = "Bye"

def main():

client\_soc = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

client\_soc.connect(ADDR)

print(f"[CONNECTED] Client connected to server at {IP} {PORT}")

connected = True

while connected:

msg = input("> ")

client\_soc.send(msg.encode(FORMAT))

if msg == DISCONNECT\_MSG:

connected = False

else :

msg = client\_soc.recv(SIZE).decode(FORMAT)

print(f"[SERVER] {msg}")

client\_soc.close()

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

main()

import socket

import threading

IP = socket.gethostbyname(socket.gethostname())

PORT = 5656

ADDR = (IP, PORT)

FORMAT = "utf-8"

SIZE = 1024

DISCONNECT\_MSG = "Bye"

def handle\_client(conn, addr):

print(f"[NEW CONNECTION] {addr} Connected ...")

connected = True

while connected:

msg = conn.recv(SIZE).decode(FORMAT)

if msg == DISCONNECT\_MSG:

connected = False

print(f"{[addr]} {msg}")

msg = f"Msg Received : {msg}"

conn.send(msg.encode(FORMAT))

conn.close()

def main():

print("[STARTING] Server is starting...")

server\_soc = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server\_soc.bind(ADDR)

server\_soc.listen()

print(f"[LISTENING] Server is listening on {IP} : {PORT}")

while True:

conn, addr = server\_soc.accept()

cl\_thread = threading.Thread(target=handle\_client, args=(conn, addr))

cl\_thread.start()

print(f"[ACTIVE CONNECTIONS] {threading.activeCount() - 1}")

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

main()