LAB – 1

- 7 AUG 2023

1. Create three programs, two of which are clients to a single server. Client1 will send a character to the server process. The server will decrement the letter to the next letter in the alphabet and send the result to client2. Client2 prints the letter it receives and then all the processes terminate.

Server.py

```
import socket


ip = "localhost" #socket.gethostname()#192.168.1.11
port = 8080
server = socket.socket()
server.bind((ip, port))
server.listen()
connection, address = server.accept()
print("connected to client1: ", address[0])
data = connection.recv(1024).decode()
connection.close()

ip = "localhost" #socket.gethostname()#192.168.1.11
port = 8081
server = socket.socket()
server.bind((ip, port))
server.listen()
connection, address = server.accept()
print("connected to client2: ", address[0])
# data = connection.recv(1024).decode()
data = chr(ord(data)+1)
connection.send(data.encode())
connection.close()
```

client1.py

```
import socket


ip = "localhost" # socket.gethostname()#192.168.1.11
port = 8080
client = socket.socket()
client.connect((ip, port))
message = "a"
client.send(message.encode())
client.close()
```

Client2.py

```
import socket


ip = "localhost" # socket.gethostname()#192.168.1.11
port = 8081
client = socket.socket()
client.connect((ip, port))
data = client.recv(1024).decode()
print("message from server: " + data)
client.close()
```

2. Write a socket program to enable client1 to send a float value to the server. The server process should increase the value of the number it receives by a power of 1.5. The server should print both the value it receives and the value that it sends. Client2 should print the value it receives from the server.

Server.py

```
import socket


ip = "localhost" #socket.gethostname()#192.168.1.11
port = 8080
server = socket.socket()
server.bind((ip, port))
server.listen()
connection, address = server.accept()
print("connected to client1: ", address[0])
data = float(connection.recv(1024).decode())
print("recieved: " + str(data))
connection.close()

ip = "localhost" #socket.gethostname()#192.168.1.11
port = 8081
server = socket.socket()
server.bind((ip, port))
server.listen()
connection, address = server.accept()
print("connected to client2: ", address[0])
# data = connection.recv(1024).decode()
data = float(data ** 1.5)
print("sending: " + str(data))
connection.send(str(data).encode())
connection.close()
```

Client1.py

```
import socket


ip = "localhost" # socket.gethostname()#192.168.1.11
port = 8080
client = socket.socket()
client.connect((ip, port))
message = "10"
client.send(message.encode())
client.close()
```

Client2.py

```
import socket


ip = "localhost" # socket.gethostname()#192.168.1.11
port = 8081
client = socket.socket()
client.connect((ip, port))
data = client.recv(1024).decode()
print("message from server: " + data)
client.close()
```

3. Send datagrams with two arrays of integers (only even numbers) to a server. The server should check the data, whether there are odd and/or fraction numbers. If it is not, the server sums the elements of each array and puts the sum in a third array that is returned to the client. If the server discovers that the arrays have erroneous then the server does not reply. A timeout period should be established by the client such that retransmission occurs after the period expires.

Server.py

```
import socket

def function(data):
    arr1 = []
    arr2 = []
    flag = False
    for i in data:
        if i=="|":
            flag = True
        elif not flag:
            if i != " ":
                arr1.append(int(i))
        else:
            if i != " ":
                arr2.append(int(i))
    for i in arr1:
        if int(i)!=i or i%2!=0:
            return ""
    for i in arr2:
        if int(i)!=i or i%2!=0:
            return ""
    if len(arr1)!=len(arr2):
        return ""
    result = []
    for i in range(len(arr1)):
        result.append(arr1[i] + arr2[i])
    return str(result)


ip = "localhost"
port = 8080
server = socket.socket()
server.bind((ip, port))
server.listen()
connection, address = server.accept()
data = connection.recv(1024).decode()
data = function(data)
if (data!=""):
    connection.send(data.encode())
connection.close()
```

Client.py

```
import socket

ip = "localhost"
port = 8080

client = socket.socket()
client.connect((ip, port))
arr1 = input("enter the first array(seperated by space \" \"): ")
arr2 = input("enter the second array(seperated by space \" \"): ")
data = arr1 + "|" + arr2
client.send(data.encode())
data = client.recv(1024).decode()
print(data)
client.close()
```

3. Implement a port scanner using socket programming. The port scanner checks a number of ports (for instance, from 1 to 1026) to see if they are open (a server is listening on that port number) or closed (a server is not listening on that port number)

main.py

```
import socket


ip = "localhost" # socket.gethostname()#192.168.1.11
result = []

client = socket.socket()
for port in range(int(input("starting port: ")), int(input("ending port: "))+1):
    try:
        client.connect((ip, port))
        client.close()
        result.append(port)
    except:
        pass
print("open ports: " + str(result))
```