Name : Md.Abdullah

ID   : IT-17015

Lab report no: 04

Name of the Lab report : Echo protocol

Objectives:

- ✓ Learn eco TCP server
- ✓ Learn eco TCP client
- ✓ Learn eco UDP server
- ✓ Learn eco UDP client

Theory:

Python's socket module provides an interface to the Berkeley sockets API. The primary socket API functions and methods in this module are:

- `socket()`
- `bind()`
- `listen()`
- `accept()`
- `connect()`
- `connect_ex()`
- `send()`
- `recv()`
- `close()`

Echo TCP server program:

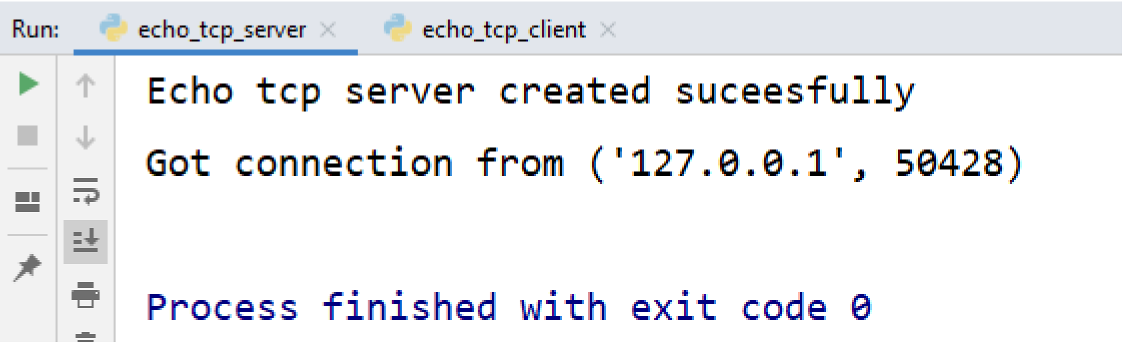
```
import socket

HOST = '127.0.0.1'
PORT = 65432
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
   print("Echo tcp server created suceesfully")
   s.bind((HOST, PORT))
   s.listen()
   conn, addr = s.accept()
   with conn:
      print('Got connection from', addr)
      while True:
         data = conn.recv(1024)
```

```
            if not data:
                break
            conn.sendall(data)
```

Echo TCP clients:

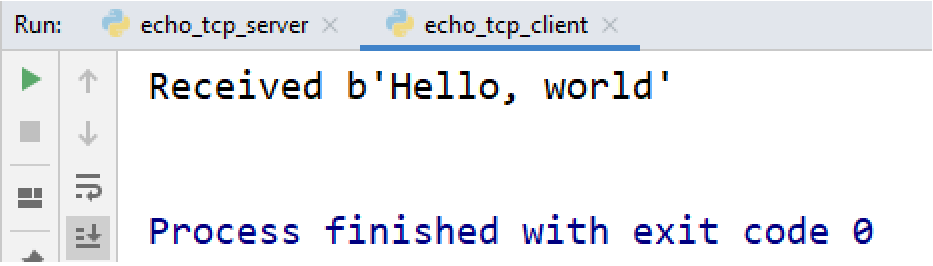
```
import socket

HOST = '127.0.0.1'  # The server's hostname or IP address
PORT = 65432        # The port used by the server

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    s.sendall(b'Hello, world')
    data = s.recv(1024)

print('Received', repr(data))
```

Running the server result:



```
Echo tcp server created suceesfully
Got connection from ('127.0.0.1', 50428)


Process finished with exit code 0
```

Running the Client result:



```
Received b'Hello, world'


Process finished with exit code 0
```

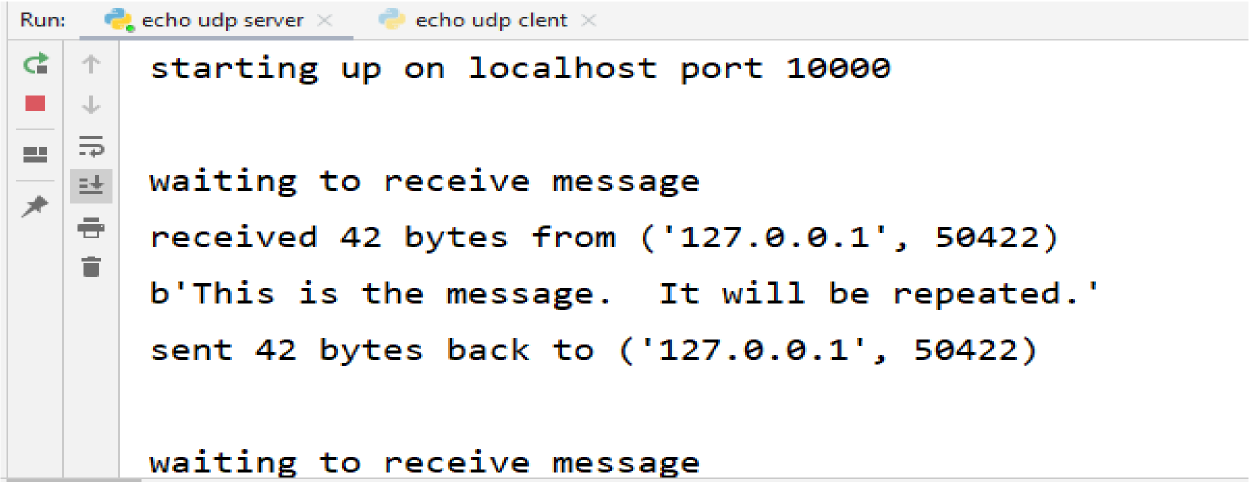
Echo UDP server:

```
import socket
import sys

# Create a TCP/IP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Bind the socket to the port
server_address = ('localhost', 10000)
print('starting up on %s port %s' % server_address)
sock.bind(server_address)
while True:
    print('\nwaiting to receive message')
    data, address = sock.recvfrom(4096)

    print('received %s bytes from %s' % (len(data), address))
    print(data)

    if data:
        sent = sock.sendto(data, address)
        print('sent %s bytes back to %s' % (sent, address))
```

Echo UDP client:

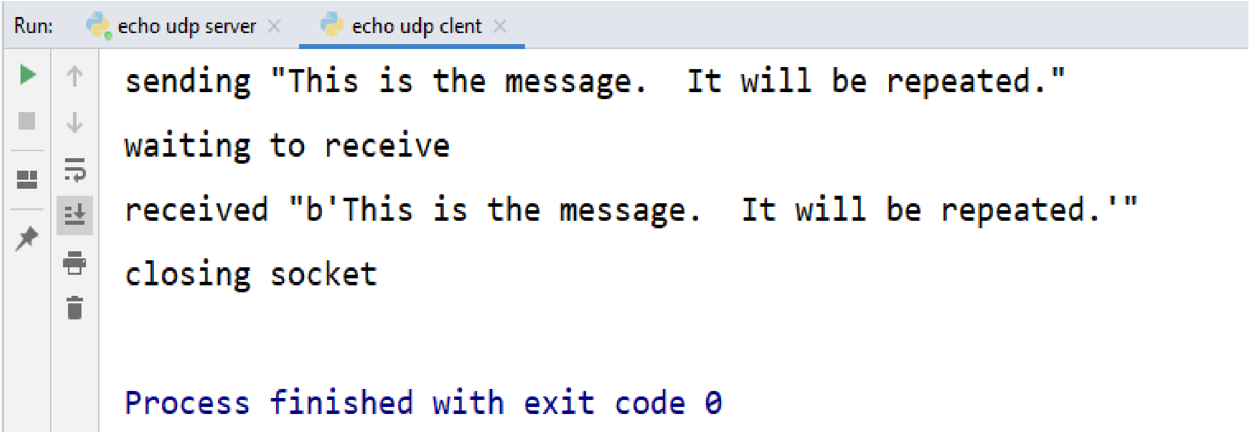
```
import socket
import sys

# Create a UDP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

server_address = ('localhost', 10000)
message = 'This is the message.  It will be repeated.'

try:

    # Send data
    print('sending "%s"' % message)
    sent = sock.sendto(message.encode('utf-8'), server_address)

    # Receive response
    print('waiting to receive')
    data, server = sock.recvfrom(4096)
    print('received "%s"' % data)

finally:
    print( 'closing socket')
    sock.close()
```

Running the server result:



```
starting up on localhost port 10000

waiting to receive message
received 42 bytes from ('127.0.0.1', 50422)
b'This is the message.  It will be repeated.'
sent 42 bytes back to ('127.0.0.1', 50422)

waiting to receive message
```

Running the client result:



```
sending "This is the message.  It will be repeated."
waiting to receive
received "b'This is the message.  It will be repeated.'"
closing socket

Process finished with exit code 0
```

Discussion:

I can successfully run all the program of eco tcp server.py , eco tcp client.py ,echo udp server.py ,echo udp client.py