

TASK-8

VU22CSEN0101456

N LOKESH

Q)Write a program to create a server that listens to port 59 using datagram sockets. Write a simple client program that requests the server for a binary file. The server should service multiple clients concurrently and send the requested files in response.

CODE :

CLIENT :

```
import socket
import threading

def file(filename):
    # Creating a UDP socket
    c = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

    # Sending the filename to the server using UDP
    c.sendto(filename.encode(), ('localhost', 59))

    # Receiving the response and the address of the server
    response, addr = c.recvfrom(4096)
    print("Connected to", addr)

    # Converting the filename to uppercase
    filename = filename.upper()

    # Writing the received content to a file
    with open(filename, 'wb') as f:
        f.write(response)

    # Printing a message indicating that the file has been received
    print('File received:', filename)

if __name__ == "__main__":
    while True:
        # Getting the filename from the user
        filename = input('Enter filename or type "quit" to exit: ')
```

```

# Checking if the user wants to exit
if filename.lower().strip() == 'exit':
    break

# Creating a new thread to handle the file retrieval
t1 = threading.Thread(target=file, args=(filename,))

# Starting the thread
t1.start()

# Waiting for the thread to finish execution
t1.join()

```

SERVER :

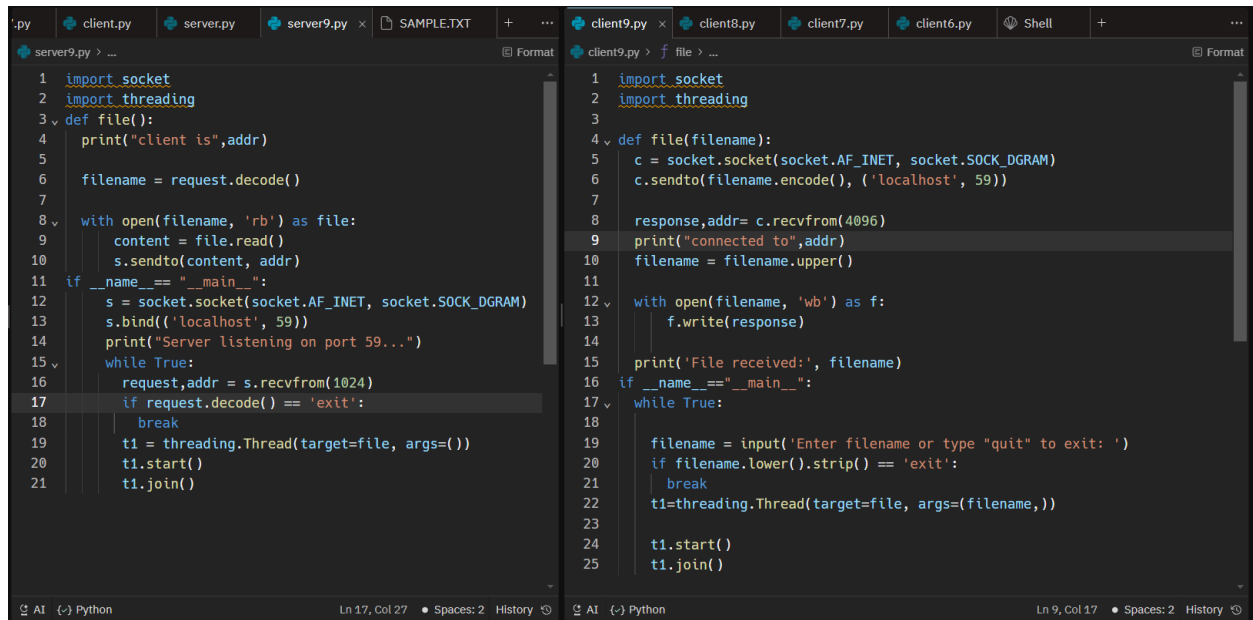
```

import socket
import threading
# Function to handle file transfer
def file():
    print("client is", addr)
    # Extract filename from the request
    filename = request.decode()

    # Open the file in binary read mode
    with open(filename, 'rb') as file:
        # Read the content of the file
        content = file.read()
        # Send the content back to the client
        s.sendto(content, addr)
# Main function
if __name__ == "__main__":
    # Create a UDP socket
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    # Bind the socket to localhost and port 59
    s.bind(('localhost', 59))
    print("Server listening on port 59...")
    # Main server loop
    while True:
        # Receive data from a client and its address
        request, addr = s.recvfrom(1024)
        # Check if the received request is to exit
        if request.decode() == 'exit':
            break
        # Create a new thread to handle the file transfer
        t1 = threading.Thread(target=file, args=())

```

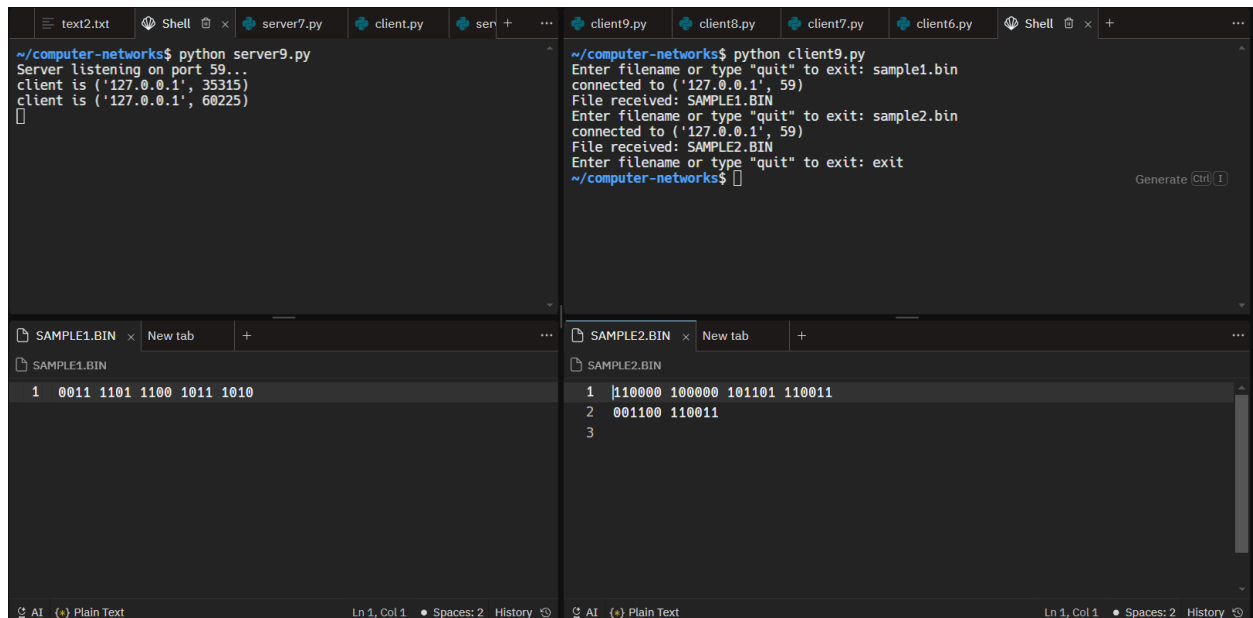
```
t1.start()
# Wait for the thread to finish execution before proceeding
t1.join()
```



```
server9.py
1 import socket
2 import threading
3 def file():
4     print("client is",addr)
5     filename = request.decode()
6
7     with open(filename, 'rb') as file:
8         content = file.read()
9         s.sendto(content, addr)
10
11 if __name__ == "__main__":
12     s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
13     s.bind(('localhost', 59))
14     print("Server listening on port 59...")
15     while True:
16         request,addr = s.recvfrom(1024)
17         if request.decode() == 'exit':
18             break
19         t1 = threading.Thread(target=file, args=())
20         t1.start()
21         t1.join()

client9.py
1 import socket
2 import threading
3
4 def file(filename):
5     c = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
6     c.sendto(filename.encode(), ('localhost', 59))
7
8     response,addr= c.recvfrom(4096)
9     print("connected to",addr)
10    filename = filename.upper()
11
12    with open(filename, 'wb') as f:
13        f.write(response)
14
15    print('File received:', filename)
16    if __name__=="__main__":
17        while True:
18
19            filename = input('Enter filename or type "quit" to exit: ')
20            if filename.lower().strip() == 'exit':
21                break
22            t1=threading.Thread(target=file, args=(filename,))
23
24            t1.start()
25            t1.join()
```

OUTPUT :



```
~/computer-networks$ python server9.py
Server listening on port 59...
client is ('127.0.0.1', 35315)
client is ('127.0.0.1', 60225)

~/computer-networks$ python client9.py
Enter filename or type "quit" to exit: sample1.bin
connected to ('127.0.0.1', 59)
File received: SAMPLE1.BIN
Enter filename or type "quit" to exit: sample2.bin
connected to ('127.0.0.1', 59)
File received: SAMPLE2.BIN
Enter filename or type "quit" to exit: exit
~/computer-networks$
```

```
SAMPLE1.BIN
1 0011 1101 1100 1011 1010

SAMPLE2.BIN
1 110000 100000 101101 110011
2 001100 110011
3
```

