Network Design: Programming Project Phase 01
Aswin Kumar Manickam 01624359

Aim:

The Aim of the project is to **transfer** a file or string between **UDP client and UDP server**.

Design:

1st step:

- The Server socket is created

2st step:

- The Client socket is created.
- The client side opens an image file 'lion.jpg' (651KB image is used in this project) and reads the file with the help of file.read().
- Buffer size used is 1024 bytes.
- File size is calculated. Number of 'loop-times' is calculated by file size/buffer size.
- Client sends the loop-times (converting into bytes in-order to transfer) and file to the Server.
- Client sends data to the server side by sending File, IP Address and port number.

3nd step:

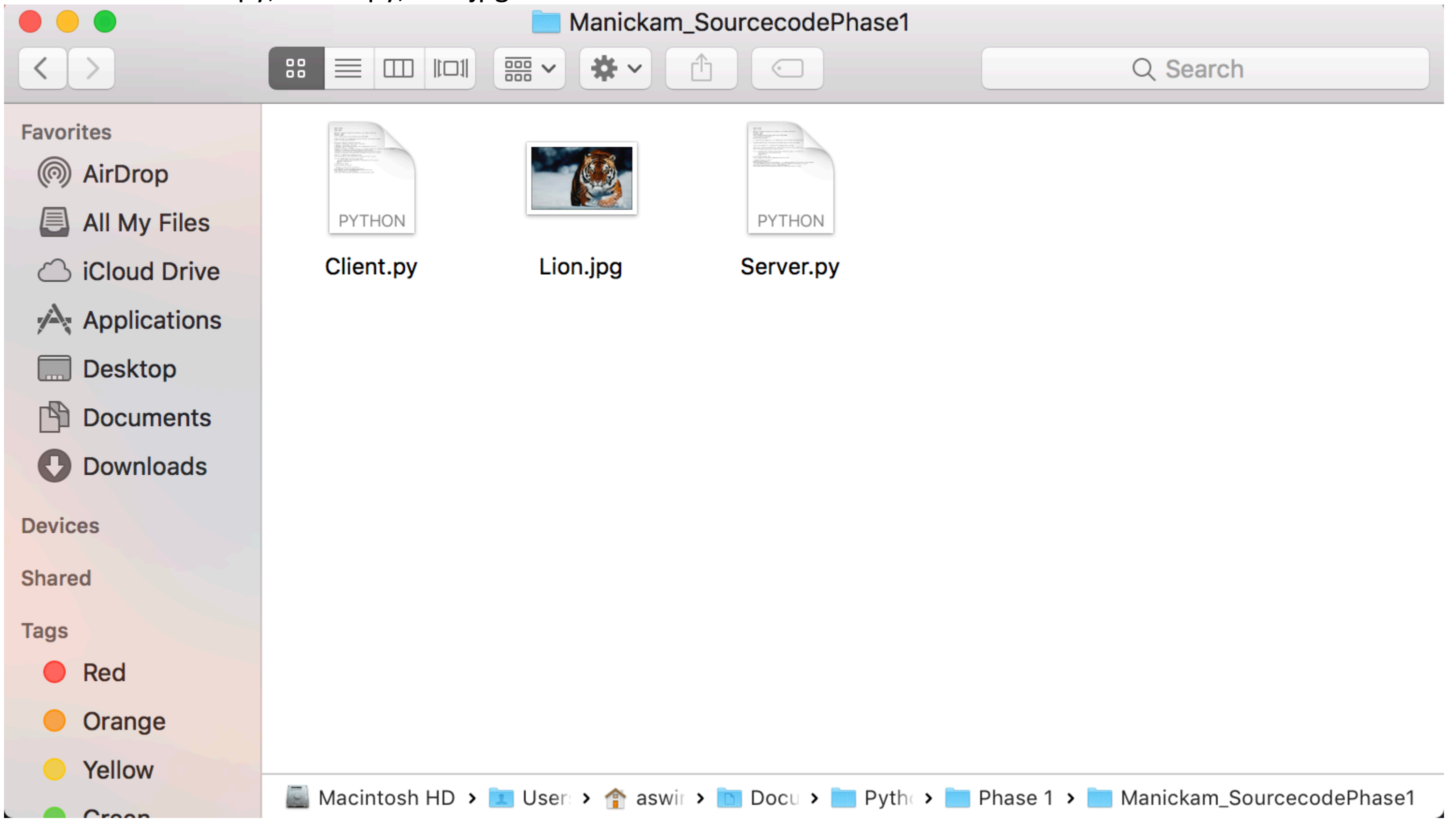
- Server Socket binds IP Address and the port number.
- Opens a new file called "Received Image.jpg" to write the transferred image from Client.
- Server receives data from the client with buffer size of 1024 bytes.
- Gets 'loop-times' from Client and changes to integer for loop purpose.
- Copy the transferred image to "Received Image.jpg" by using file.write() function, here we are using 'FOR' loop '0 to loop' times to write the entire image into new file.
- File has been received, now you can see one more file called received file.jpg in the folder with the transferred image.
- Now send a message to from Server to Client in order to check data transfer happens both directions.

4rd step:

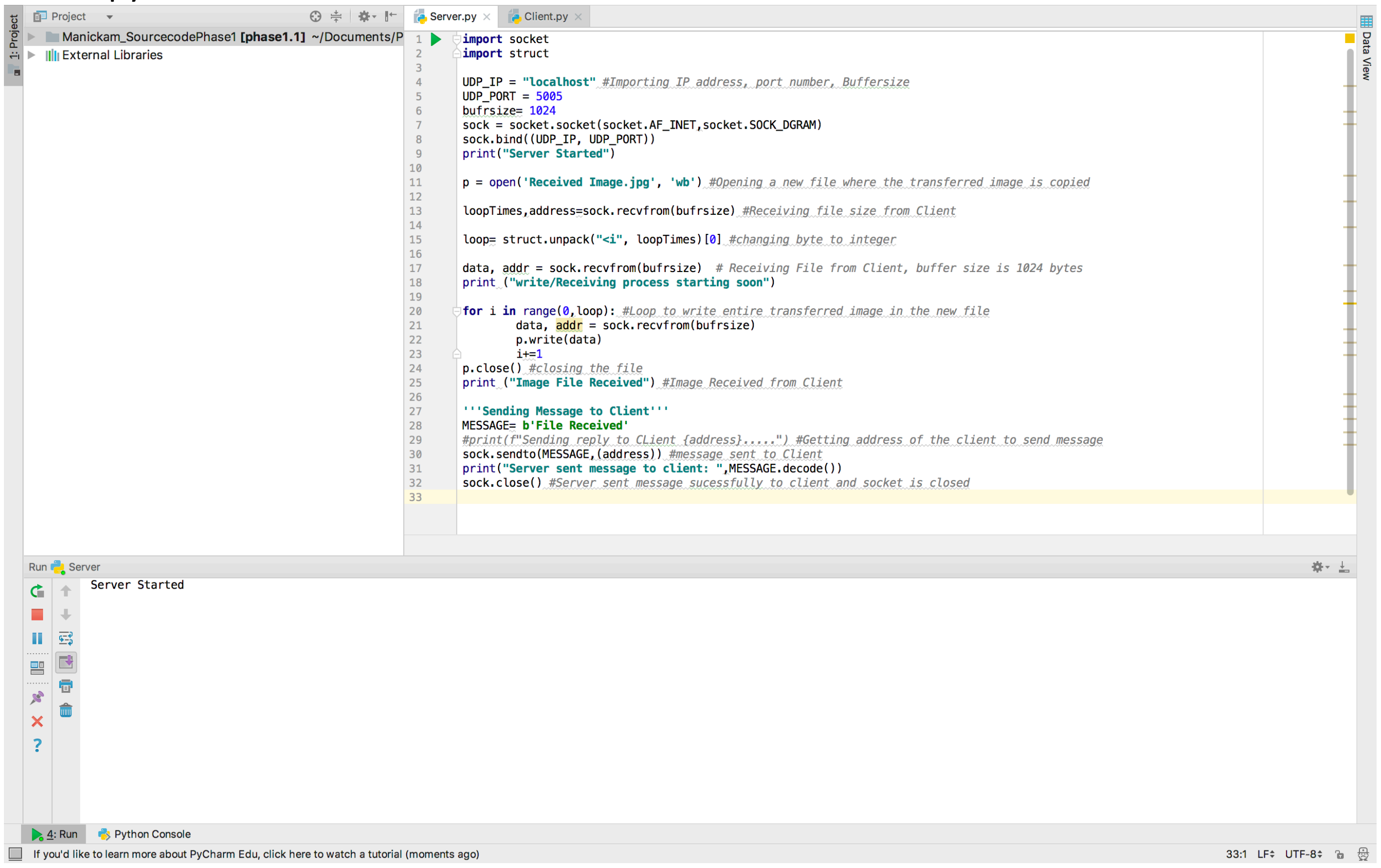
- Message is received in the client side from Server.
- Receive data and print.
- Check in console to verify that the data has been received from the Server.

Execution of the program:

Folder has Server.py, Client.py, lion.jpg for transfer:



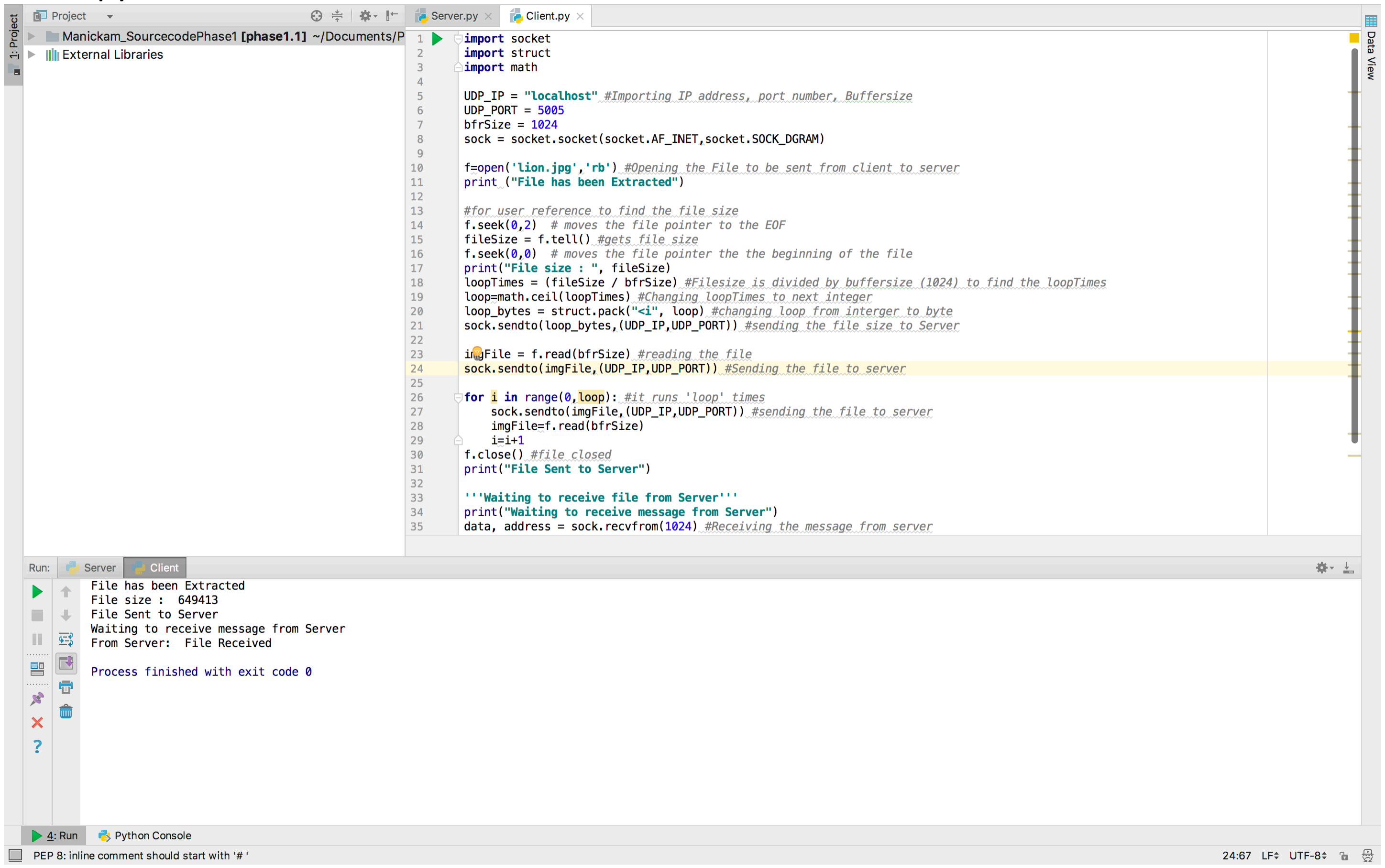
Server.py is executed first:


Project
Manickam_SourcecodePhase1 [phase1.1] ~/Documents/P
External Libraries
Server.py
Client.py
Run Server
4: Run
Python Console
If you'd like to learn more about PyCharm Edu, click here to watch a tutorial (moments ago)
33:1 LF UTF-8

```python
import socket
import struct

UDP_IP = "localhost" #Importing IP address, port number, Buffersize
UDP_PORT = 5005
bufrsize= 1024
sock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
sock.bind((UDP_IP, UDP_PORT))
print("Server Started")

p = open('Received Image.jpg', 'wb') #Opening a new file where the transferred image is copied

loopTimes,address=sock.recvfrom(bufrsize) #Receiving file size from Client

loop= struct.unpack("<i", loopTimes)[0] #changing byte to integer

data, addr = sock.recvfrom(bufrsize)  # Receiving File from Client, buffer size is 1024 bytes
print ("write/Receiving process starting soon")

for i in range(0,loop): #Loop to write entire transferred image in the new file
        data, addr = sock.recvfrom(bufrsize)
        p.write(data)
        i+=1
p.close() #closing the file
print ("Image File Received") #Image Received from Client

'''Sending Message to Client'''
MESSAGE= b'File Received'
#print(f"Sending reply to CLient {address}.....") #Getting address of the client to send message
sock.sendto(MESSAGE,(address)) #message sent to Client
print("Server sent message to client: ",MESSAGE.decode())
sock.close() #Server sent message sucessfully to client and socket is closed
```

```
Server Started
```

Client.py is executed:

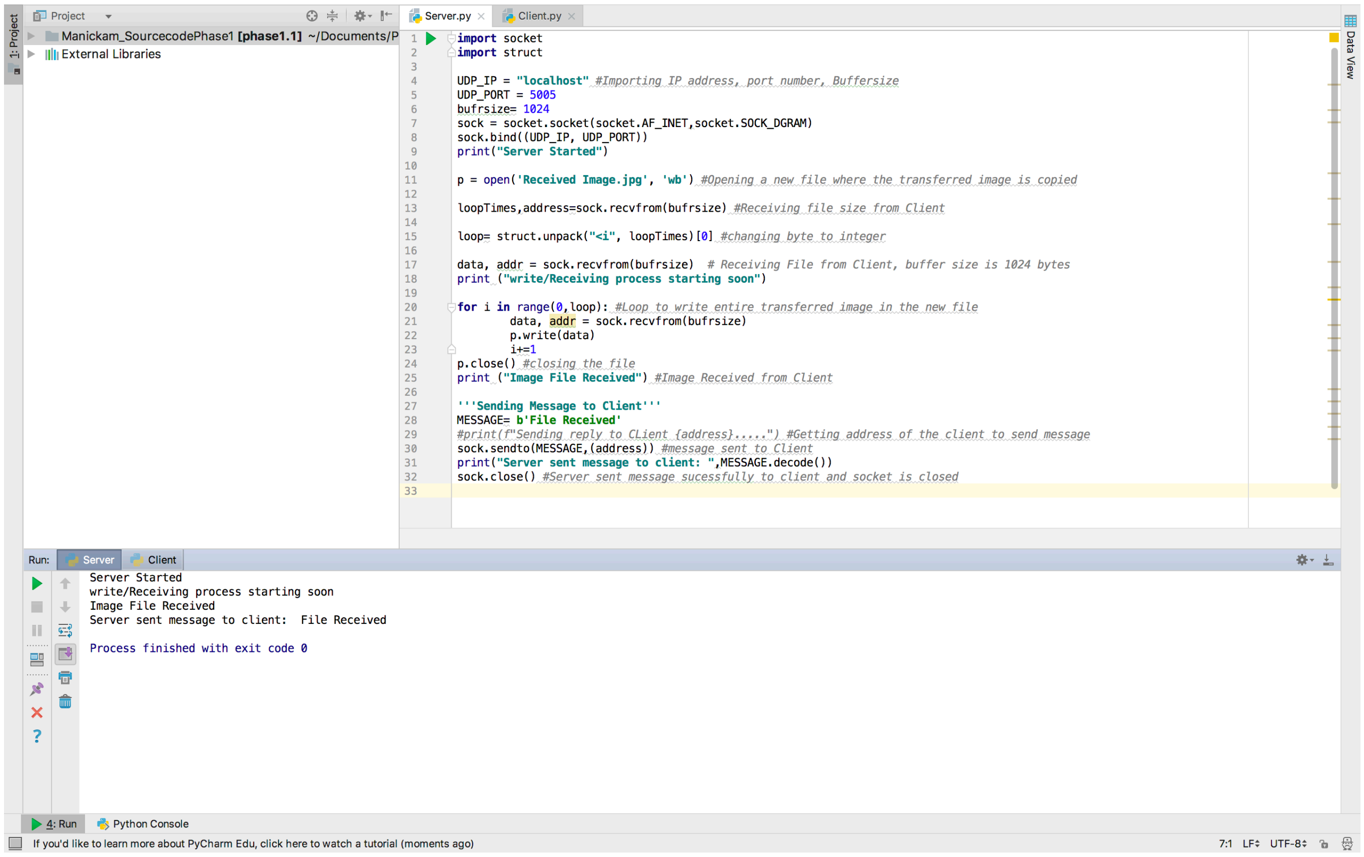


```python
import socket
import struct
import math

UDP_IP = "localhost" #Importing IP address, port number, Buffersize
UDP_PORT = 5005
bfrSize = 1024
sock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)

f=open('lion.jpg','rb') #Opening the File to be sent from client to server
print ("File has been Extracted")

#for user reference to find the file size
f.seek(0,2)  # moves the file pointer to the EOF
fileSize = f.tell() #gets file size
f.seek(0,0)  # moves the file pointer the the beginning of the file
print("File size : ", fileSize)
loopTimes = (fileSize / bfrSize) #Filesize is divided by buffersize (1024) to find the loopTimes
loop=math.ceil(loopTimes) #Changing loopTimes to next integer
loop_bytes = struct.pack("<i", loop) #changing loop from interger to byte
sock.sendto(loop_bytes,(UDP_IP,UDP_PORT)) #sending the file size to Server

imgFile = f.read(bfrSize) #reading the file
sock.sendto(imgFile,(UDP_IP,UDP_PORT)) #Sending the file to server

for i in range(0,loop): #it runs 'loop' times
    sock.sendto(imgFile,(UDP_IP,UDP_PORT)) #sending the file to server
    imgFile=f.read(bfrSize)
    i=i+1
f.close() #file closed
print("File Sent to Server")

'''Waiting to receive file from Server'''
print("Waiting to receive message from Server")
data, address = sock.recvfrom(1024) #Receiving the message from server
```

```
File has been Extracted
File size :  649413
File Sent to Server
Waiting to receive message from Server
From Server:  File Received

Process finished with exit code 0
```

**File transferred from Client to server and also the Message has been received from server to Client side

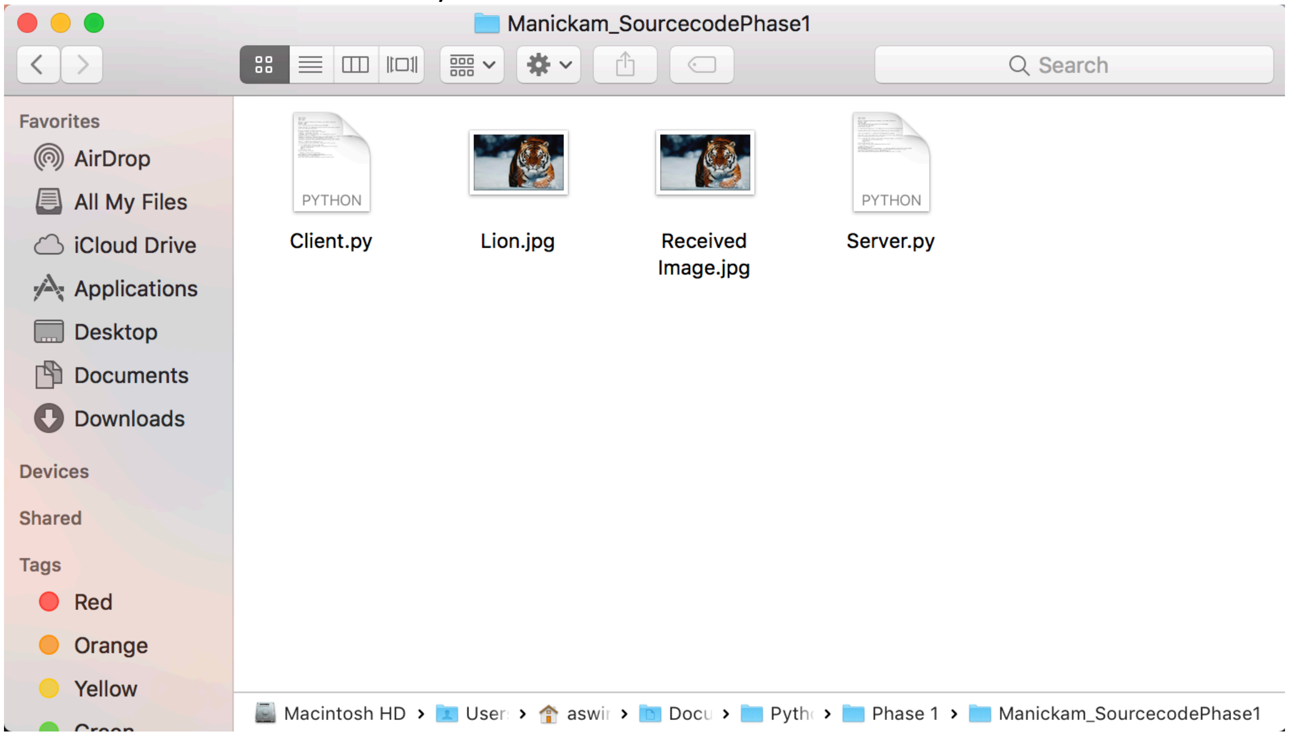
Server console:


Project
Manickam_SourcecodePhase1 [phase1.1] ~/Documents/P
External Libraries
Server.py
Client.py
Run:
Server
Client
4: Run
Python Console
If you'd like to learn more about PyCharm Edu, click here to watch a tutorial (moments ago)
7:1 LF÷ UTF-8÷

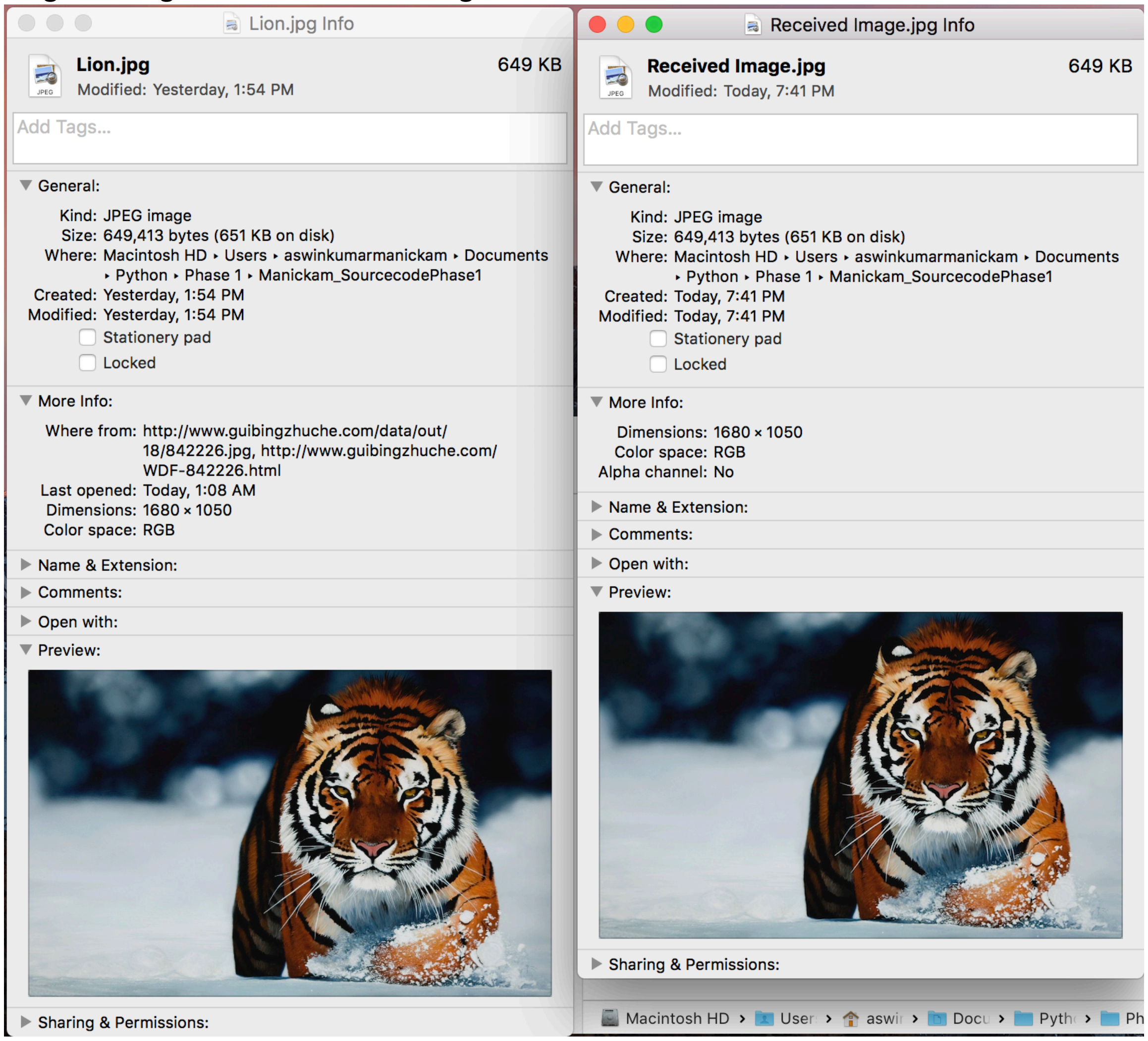
```python
import socket
import struct

UDP_IP = "localhost" #Importing IP address, port number, Buffersize
UDP_PORT = 5005
bufrsize= 1024
sock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
sock.bind((UDP_IP, UDP_PORT))
print("Server Started")

p = open('Received Image.jpg', 'wb') #Opening a new file where the transferred image is copied

loopTimes,address=sock.recvfrom(bufrsize) #Receiving file size from Client

loop= struct.unpack("<i", loopTimes)[0] #changing byte to integer

data, addr = sock.recvfrom(bufrsize)  # Receiving File from Client, buffer size is 1024 bytes
print ("write/Receiving process starting soon")

for i in range(0,loop): #Loop to write entire transferred image in the new file
        data, addr = sock.recvfrom(bufrsize)
        p.write(data)
        i+=1
p.close() #closing the file
print ("Image File Received") #Image Received from Client

'''Sending Message to Client'''
MESSAGE= b'File Received'
#print(f"Sending reply to CLient {address}.....") #Getting address of the client to send message
sock.sendto(MESSAGE,(address)) #message sent to Client
print("Server sent message to client: ",MESSAGE.decode())
sock.close() #Server sent message sucessfully to client and socket is closed

```

```
Server Started
write/Receiving process starting soon
Image File Received
Server sent message to client:  File Received

Process finished with exit code 0
```

File transferred from Client to Server, also check "received image.jpg" in the file folder to confirm that the file has been transferred successfully:



Original image and Transferred image same size:



Lion.jpg 649 KB
Modified: Yesterday, 1:54 PM

General:
Kind: JPEG image
Size: 649,413 bytes (651 KB on disk)
Where: Macintosh HD ‣ Users ‣ aswinkumarmanickam ‣ Documents ‣ Python ‣ Phase 1 ‣ Manickam_SourcecodePhase1
Created: Yesterday, 1:54 PM
Modified: Yesterday, 1:54 PM
Stationery pad
Locked

More Info:
Where from: http://www.guibingzhuche.com/data/out/18/842226.jpg, http://www.guibingzhuche.com/WDF-842226.html
Last opened: Today, 1:08 AM
Dimensions: 1680 × 1050
Color space: RGB

Received Image.jpg 649 KB
Modified: Today, 7:41 PM

General:
Kind: JPEG image
Size: 649,413 bytes (651 KB on disk)
Where: Macintosh HD ‣ Users ‣ aswinkumarmanickam ‣ Documents ‣ Python ‣ Phase 1 ‣ Manickam_SourcecodePhase1
Created: Today, 7:41 PM
Modified: Today, 7:41 PM
Stationery pad
Locked

More Info:
Dimensions: 1680 × 1050
Color space: RGB
Alpha channel: No

Result:

Client and Server communication using UDP has been executed.