**PRACTICAL NO.02**

**Multithreaded Server( server.py)**

import socket

from threading import Thread

class ClientThread(Thread):

def \_\_init\_\_(self,ip,port,sock):

Thread.\_\_init\_\_(self)

self.ip = ip

self.port = port

self.sock = sock

print (" New thread started for "+ip+":"+str(port))

def run(self):

filename='folder/abcd.txt'

f = open(filename,'rb')

while True:

l = f.read(1024)

while (l):

self.sock.send(l)

l = f.read(1024)

if not l:

f.close()

self.sock.close()

break

sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

sock.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1)

sock.bind(("127.0.0.1", 12341))

threads = []

while True:

sock.listen(5)

print ("Waiting for incoming connections...")

(conn, (ip,port)) = sock.accept()

print ('Got connection from ', (ip,port))

newthread = ClientThread(ip,port,conn)

newthread.start()

threads.append(newthread)

for t in threads:

t.join()

**Client( client.py)**

import socket

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

s.connect(("127.0.0.1", 12341))

f=open('received\_file.txt', 'wb')

print ('file opened')

while True:

print('receiving data...')

data = s.recv(1024)

print('data=%s', (data))

if not data:

f.close()

print ('file closed')

break

# write data to a file

f.write(data)

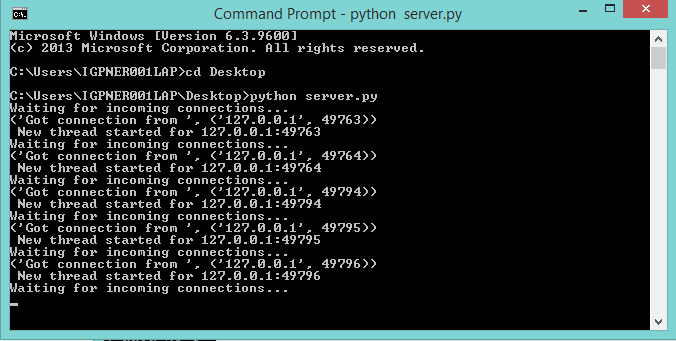
print('Successfully get the file')

s.close()

print('connection closed')

Output:

server.py



Client.py

