COMPUTER NETWORK

***Lab* 9**

Zia Ur Rehman

1802034

Server

```
import pickle
import socket
from os import listdir
from os.path import isfile, join
import os
path = 'files/'
onlyfiles = [f for f in listdir(path) if isfile(join(path, f))]

# from commonthread import commonThread

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
client_req = []
client_resp = []
udp_host = "192.168.100.50"
udp_port = 12345


sock.bind((udp_host, udp_port))


def send_file_chunk(addr):
    CHUNK_SIZE = 100
    offset=0
    hex_string="0x0012"
    client_resp.clear()
    f = open(join(path,onlyfiles[0]), 'r')
    chunk = f.read(CHUNK_SIZE)
    while chunk:
        client_resp.append(hex_string)
        client_resp.append(hex(offset))
        client_resp.append(chunk)
        print("file sending ....")
        sock.sendto(pickle.dumps(client_resp), (addr[0], addr[1]))
        client_resp.clear()
        chunk=f.read(CHUNK_SIZE) #read the next chunk
        offset+=1

    #loop until the chunk is empty (the file is exhausted)
    print("file sending complete....")

    f.close()

def file_list(addr):
    hex_string = "0x0010"
    client_resp = onlyfiles.copy()
```

```
    client_resp.insert(0, hex(len(onlyfiles)))
    client_resp.insert(0, hex_string)
    sock.sendto(pickle.dumps(client_resp), (addr[0], addr[1]))

def send_file(addr):
    hex_string="0x0011"
    client_resp.append(hex_string)
    client_resp.append(client_req[1])
    status=os.stat(join(path,client_req[1]))
    client_resp.append(status.st_size)
    print(client_resp)
    sock.sendto(pickle.dumps(client_resp),(addr[0],addr[1]))
    send_file_chunk(addr)
while True:
    print("Waiting for client")
    data, addr = sock.recvfrom(1024)
    print("Receied Messages:", pickle.loads(data), "form ", addr)
    sock.sendto(pickle.dumps("Yes"), (addr[0], addr[1])) #ack
    data, addr = sock.recvfrom(1024)#file list req
    if(pickle.loads(data) == "0x0000"):
        file_list(addr)
    data,addr=sock.recvfrom(1024)#file req
    client_req=pickle.loads(data)
    print(client_req)
    if(client_req[0]=="0x0001" and client_req[1] in onlyfiles):
        send_file(addr)
    else:
        message=['0',"File Does not Exits"]
        sock.sendto(pickle.dumps("File Does not Exits"), (addr[0], addr[1]))
```

Client

```
import socket
import pickle
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

#
# socket.gethostname()
udp_host = "192.168.100.50"
udp_port = 12345
req_file=[]

msg="Connection"
sock.sendto(pickle.dumps(msg), (udp_host, udp_port))
data,addr=sock.recvfrom(1024)
print("Receied Messages:", pickle.loads(data), "form ",addr)
# Server Responded Yes
```

```
hex_string="0x0000"
sock.sendto(pickle.dumps(hex_string), (udp_host, udp_port))


data,addr=sock.recvfrom(1024)
message=pickle.loads(data)
print(message)

# file req
req_file.append("0x0001")
req_file.append(message[2])
print(req_file)
sock.sendto(pickle.dumps(req_file),(udp_host,udp_port))

# file response
data,addr=sock.recvfrom(1024)
message=pickle.loads(data)
if(message[0]=='0'):
    print(message[1])
    input("Press Enter to Exit")
else:
    end=message[2]/100
    file=""
    while(end>=0):
        data,addr=sock.recvfrom(1024)
        message=pickle.loads(data)
        file=file+message[2]
        end-=1
    print(file)
    input("Press Enter to Exit")
# close the connection

input("Enter")
```

Output

```
Waiting for client
Receied Messages: Connection form  ('192.168.100.50', 51300)

['0x0001', 'Deutsch.txt']
['0x0011', 'Deutsch.txt', 601]
File requested
file sending ....
file sending ....
file sending ....
file sending ....
file sending ....
file sending ....
file sending ....
file sending complete....
Waiting for client
```

```
Receied Messages: Yes form  ('192.168.100.50', 12345)
['0x0010', '0x5', 'Deutsch.txt', 'Filipino.txt', 'Lorem.txt'
, 'NederLands.txt', 'Norsk.txt']
['0x0001', 'Deutsch.txt']
Es ist ein lang erwiesener Fakt, dass ein Leser vom Text abg
elenkt wird, wenn er sich ein Layout ansieht. Der Punkt, Lor
em Ipsum zu nutzen, ist, dass es mehr oder weniger die norma
le Anordnung von Buchstaben darstellt und somit nach lesbare
r Sprache aussieht. Viele Desktop Publisher und Webeditoren
nutzen mittlerweile Lorem Ipsum als den Standardtext, auch d
ie Suche im Internet nach "lorem ipsum" macht viele Webseite
n sichtbar, wo diese noch immer vorkommen. Mittlerweile gibt
 es mehrere Versionen des Lorem Ipsum, einige zufÃ¤llig, and
ere bewusst (beeinflusst von Witz und des eigenen Geschmacks
)
Press Enter to Exit
```