COSC264 Assignment 2021

Logan Lee - 26029766

August 19, 2021

Contents

1 Server

```
import socket
from datetime import datetime


"""Server program for COSC264 Assignment
    Author: Logan Lee
    Username: lpl29
    Student Id: 26029766
    """

HOST = '127.0.0.1'
port = None

def is_int(s):
    try:
        int(s)
        return True
    except ValueError as er:
        print('Value entered should be an integer')
        return False

def fileresponse(connection, status, dataLen=None,
    fileData=None):
    rspnse = bytearray(8)
    rspnse[0] = 0x49
    rspnse[1] = 0x7E
    rspnse[2] = 0x02
    rspnse[3] = status
    if status == 0:
        rspnse[4] = 0x00
        rspnse[5] = 0x00
        rspnse[6] = 0x00
        rspnse[7] = 0x00
    else:
        rspnse[4] = dataLen >> 24
        rspnse[5] = (dataLen & 0x00FF0000) >> 16
        rspnse[6] = (dataLen & 0x0000FF00) >> 8
        rspnse[7] = (dataLen & 0x000000FF)

        for i in range(dataLen):
            rspnse.append(fileData[i])

    connection.sendall(rspnse)
```

```
port = input("Enter port: ")
if is_int(port):
    port = int(port)
    if port < 1024 or port > 64000:
        print('Integer should be between 1024 and 64000')
        print('Quitting...')
        quit()

try:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.settimeout(1)
    s.bind((HOST, port))
except socket.error as e:
    print(e)
    print('Quitting...')
    s.close()
    quit()
except socket.timeout:
    print('Socket creation or bind time out')
    print('Quitting...')
    s.close()
    quit()

try:
    s.listen()
except socket.error:
    print('Connection request error')
    print('Quitting...')
    s.close()
    quit()

while True:
    try:
        conn, addr = s.accept()
        conn.settimeout(1)
        time = datetime.now().strftime("%H:%M:%S")
        print()
        print('Connected to {}, port {} at {}'.format(
            addr[0], addr[1], time))
        print()
    except socket.timeout:
        continue

    try:
```

```
        data = conn.recv(5)
        conn.settimeout(1)
        if data[0] != 0x49 or data[1] != 0x7E:
            print('MagicNo incorrect')
            print('Retrying..')
            print()
            continue
        if data[2] != 0x01:
            print('Type value incorrect')
            print('Retrying..')
            print()
            continue
        filennameLen= data[3]*2**8 + data[4]
        if filennameLen > 1024:
            print('File name too long')
            print('Retrying..')
            print()
            continue
        elif filennameLen< 1:
            print('File name too short')
            print('Retrying..')
            print()
            continue
    except socket.timeout:
        print('Timed out, took to long to recieve data')
        print('Retrying..')
        print()
        conn.close()
        addr = None
        continue

    try:
        data = conn.recv(filennameLen)
        conn.settimeout(1)
        filename = bytearray()
        for i in range(filennameLen):
            filename.append(data[i])
    except IndexError:
        fileresponse(conn, 0)
        conn.close()
    except socket.timeout:
        fileresponse(conn, 0)
        conn.close()

    try:
        data_len = 0
```

```
            file = open(filename.decode('ascii'), 'r')
            lines = file.readlines()
            fileData = bytearray()
            for line in lines:
                byte_line = line.encode('utf-8')
                for j in range(len(byte_line)):
                    fileData.append(byte_line[j])
                    data_len += 1

            fileresponse(conn, 1, data_len, fileData)
            file.close()
            conn.close()
            print('Data from ' + filename.decode('ascii') + '
                sent to client in ' + \
                  str(data_len) + ' bytes (excluding header).
                      ')
            print()

        except FileNotFoundError as er:
            fileresponse(conn, 0)
            print('File ' + filename.decode('ascii') + ' not
                found')
            print('Retrying...')
            print()
            conn.close()
            continue
```

2 Client

```
import socket

"""Client program for COSC264 Assignment
   Author: Logan Lee
   Username: lpl29
   Student Id: 26029766
   """

port = None

def addr_check(addr):
    try:
        addrinfo = socket.getaddrinfo(addr, port)
        return addrinfo[0][4][0]
    except socket.error:
        print('Address wrong or not correctly formatted')
        print('Quitting...')
        quit()

def port_check(port):
    try:
        port = int(port)
        if port < 1024 or port > 64000:
            print('Integer should be between 1024 and
                64000')
            print('Quitting...')
            quit()
        else:
            return port
    except ValueError as er:
        print('Value entered should be an integer')
        print('Quitting...')
        quit()

def file_check(file):
    try:
        o_file = open(file)
        print('File with this name already found in
            directory')
        print('Quitting...')
        o_file.close()
        quit()
    except FileNotFoundError:
```

```
        return file

addr = None
port = None
file = None

try:
    addr, port, file = map(str,input('Enter address, port
        and file seperated by spaces:\n').split())
except ValueError as er:
    print(er)
    print('Quitting...')
    quit()

addr = addr_check(addr)
port = port_check(port)
file = file_check(file)

try:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.settimeout(1)
except socket.error:
    print('Failed to create socket')
    print('Quitting...')
    s.close()
    quit()
try:
    s.connect((addr, port))
except socket.error:
    print('Connection to server timed out')
    print('Quitting...')
    s.close()
    quit()

file_request = bytearray(5)
file_request[0] = 0x49
file_request[1] = 0x7E
file_request[2] = 0x01
filenameLen = len(file)
if filenameLen < 1 and filenameLen > 1024:
    print('Filename Length is invlaid')
    print('Quitting...')
    s.close()
    quit()
else:
    file_request[3] = filenameLen >> 8
```

```
    file_request[4] = filenameLen & 0x00FF
    filename_bytes = bytes(file, 'utf-8')
    for i in range(filenameLen):
        file_request.append(filename_bytes[i])
    s.sendall(file_request)

try:
    data = s.recv(8)
    s.settimeout(1)
    if data[0] != 0x49 or data[1] != 0x7E:
        print('Incorrect MagicNo')
        print('Quitting...')
        s.close()
        quit()
    if data[2] != 0x02:
        print('Incorrect Type')
        print('Quitting...')
        s.close()
        quit()
    filennameLen= data[3]*2**8 + data[4]
    if data[3] == 0:
        print('File unable to be opened on server side')
        print('Quitting...')
        s.close()
        quit()
    file_data_len = data[4]*2**(24) + data[5]*2**16 +
       data[6]*2**8 + data[7]
    if file_data_len < 1:
        print('File has no data')
        print('Quitting...')
        s.close()
        quit()

except socket.timeout:
    print('Took to long to gather response file header')
    print('Quitting...')
    s.close()
    quit()

try:
    w_file = open(file, 'w')
    byte_count = 0


    data = s.recv(4096)
    s.settimeout(1)
```

```
    while data:
        file_data = bytearray()
        for i in range(len(data)):
            file_data.append(data[i])
            byte_count += 1
        w_file.write(file_data.decode('ascii'))
        data = s.recv(4096)
        s.settimeout(1)

    if byte_count < file_data_len:
        print('Error:')
        print('Bytes written to local file is less than
            file length')
        print('Quitting...')
        s.close()
        w_file.close()
        quit()
    elif byte_count > file_data_len:
        print('Error:')
        print('Bytes written to local file is more than
            file length')
        print('Quitting...')
        s.close()
        w_file.close()
        quit()

    print()
    print('Received data written to local file
        sucessfully')
    print('Total bytes written: ' + str(byte_count))
    print('Completed and exitting...')
    w_file.close()
    s.close()
    quit()

except socket.timeout:
    print('Took to long to gather response file data')
    print('Quitting...')
    s.close()
    w_file.close()
    quit()
except FileExistsError:
    print('The given file already exists')
    print('Quitting...')
    s.close()
    quit()
```