## TCPserver.py

from socket import \*

import sys

port = 10000

IP\_address = gethostbyname(gethostname())

serverSocket = socket(AF\_INET, SOCK\_STREAM)

server\_address = (IP\_address,port)

serverSocket.bind(server\_address)

serverSocket.listen(1)

print('The server ready! ')

def find\_leap\_year(y):

year = list(map(int, y.split(",")))

for i in year:

if (i % 4 == 0): # leap years are divisible by 4,100,and 400, or 4 but not 100.

if (i % 100 == 0): # non-Leap years are not divisible by 4 or is divisible by 4 and 100 but not 400.

if (i % 400 == 0):

print ('Year {} is a leap year!'.format(i))

else:

print ('Year {} is not a leap year!'.format(i))

else:

print ('Year {} is a leap year!'.format(i))

else:

print ('Year {} is not a leap year!'.format(i))

return "That's all"

connectionSocket, address = serverSocket.accept() # listen and accept port from client

while True:

y = connectionSocket.recv(1024).decode('utf-8')

try:

if (y == 'exit'):

print('The client wants to exit')

resp = "Exit"

connectionSocket.send(resp.encode('utf-8'))

sys.exit()

else:

print('Received the following: {!r}'.format(y))

except ValueError:

print('Received invalid values. Please try again.')

else:

try:

resp = find\_leap\_year(y)

connectionSocket.send(resp.encode('utf-8'))

except ValueError:

resp = "Received invalid values. Please try again."

connectionSocket.send(resp.encode('utf-8'))

else:

print('Success. Send the results back to the client!')

serverSocket.close()

## TCPclient.py

from socket import \*

import sys

port = 10000

IP\_adress = gethostbyname(gethostname())

clientSocket = socket(AF\_INET, SOCK\_STREAM)

client\_address =(IP\_adress,port)

clientSocket.connect(client\_address)

while True:

year = input('Enter a text file containing the test cases. Type exit to close: ')

if (year != "exit"):

with open(year, 'r') as file:

year = file.read()

clientSocket.send(year.encode('utf-8'))

resp = clientSocket.recv(1024).decode('utf-8')

print('Received from server:')

if(resp == "Exit"):

print(resp)

break

sys.exit()

else:

print(resp)

clientSocket.close()

## UDPserver.py

from socket import \*

import sys

port = 12000

print('IP Address: ', gethostbyname(gethostname()))

serverSocket = socket(AF\_INET, SOCK\_DGRAM)

serverSocket.bind((gethostbyname(gethostname()), port))

print("The server is ready!")

def find\_leap\_year(y):

year = list(map(int, y.split(",")))

for i in year:

if (i % 4 == 0): # leap years are divisible by 4,100,and 400, or 4 but not 100.

if (i % 100 == 0): # non-Leap years are not divisible by 4 or is divisible by 4 and 100 but not 400.

if (i % 400 == 0):

print ('Year {} is a leap year!'.format(i))

else:

print ('Year {} is not a leap year!'.format(i))

else:

print ('Year {} is a leap year!'.format(i))

else:

print ('Year {} is not a leap year!'.format(i))

return "That's all"

while True:

y, clientAddress = serverSocket.recvfrom(2048)

y = y.decode('utf-8')

try:

if ((y == 'exit')):

print('The client wants to exit')

resp = "Exit"

serverSocket.sendto(resp.encode('utf-8'),clientAddress)

sys.exit()

else:

print('Received the following years: {!r}'.format(y))

except ValueError:

print('Invalid values. Please try again.')

serverSocket.sendto("Invalid values. Please try again.",clientAddress)

else:

try:

resp = find\_leap\_year(y)

print(resp)

serverSocket.sendto(resp.encode('utf-8'),clientAddress)

except ValueError:

resp = "Invalid values. Please try again."

serverSocket.sendto(resp.encode('utf-8'), clientAddress)

else:

print('Success. Send the result back to the client!')

serverSocket.close()

## UDPclient.py

from socket import \*

port = 12000

IP\_address = gethostbyname(gethostname())

clientSocket = socket(AF\_INET, SOCK\_DGRAM) #create udp client socket

serverAddress =(IP\_address,port)

while True:

year = input('Enter a text file containing the test cases. Type exit to close: ')

if (year != "exit"):

with open(year, 'r') as file:

year = file.read()

clientSocket.sendto(year.encode('utf-8'), serverAddress)

resp, serverAddress = clientSocket.recvfrom(2048)

resp = resp.decode('utf-8')

print('Received from server:')

if(resp == "Exit"):

print(resp)

break

sys.exit()

else:

print(resp)

clientSocket.close()