**Server CODE**

**CODE:**

import socket

#Connect to clients one by one as instructed by program

client=[] #An array of all the the clients

s=['s1','s2','s3'] #Sockets of every different client(3 in this Case).

order=0

i=0

count=0 #some variables declared to be used in iteration.

loc\_dur=[] #An array containing location &

#time duration required to rech the destination.

# Establishing connections of all clients using iteration

def accept\_connections():

for i in range(0,3):

s[i]=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM) #Socket creation

s[i].bind((socket.gethostname(),2000+i)) #binding socket to the port

s[i].listen(5) #putting the socket into listening mode

print(f"Waiting for Truck {i+1} to connect......")

clt,addr=s[i].accept()

# .accept() functions waits for clients to connect

client.append(clt) #adding client address to

print(f"Connection established to Truck {i+1}")

print("Connected to all 3 Clients!!")

# append delivery information to a output file

def write\_output():

global f

f = open("Output.txt", "a+") # a+ Open for reading and appending

#(writing at end of file).

#The file is created if it does not exist.

# input Delivery locations and time duration to complete delivery

# \*seperated by a ' '

def take\_input():

global loc\_dur

global n

n=int(input("Enter Number of locations:"))

for i in range(n):

loc\_dur.append(input().split(" "))

def assigning\_task():

global order,i,count

global loc\_dur

global client

delay1=0

delay2=0

delay3=0

while(i >=0):

if(count<n and delay1==0):

print(f"{loc\_dur[0][0]} Truck 1")

delay1=int(loc\_dur[0][1])

client[0].sendall(bytes(str(loc\_dur[0][0]+" Truck 1"),"ascii"))

f.write(str(loc\_dur[0][0]+" Truck 1\n")) #Writes the info. of Truck & Order in a seperate output file

loc\_dur=loc\_dur[1:] #Keeps deleting the entry from the array, which is assigned

count+=1 #Keeps a count of no. of orders assigned to a particular Truck

if(count<n and delay2==0):

print(f"{loc\_dur[0][0]} Truck 2")

delay2=int(loc\_dur[0][1])

client[1].sendall(bytes(str(loc\_dur[0][0]+" Truck 2"),"ascii"))

f.write(str(loc\_dur[0][0]+" Truck 2\n"))

loc\_dur=loc\_dur[1:]

count+=1

if(count<n and delay3==0):

print(f"{loc\_dur[0][0]} Truck 3")

delay3=int(loc\_dur[0][1])

client[2].sendall(bytes(str(loc\_dur[0][0]+" Truck 3"),"ascii"))

f.write(str(loc\_dur[0][0]+" Truck 3\n"))

loc\_dur=loc\_dur[1:]

count+=1

# Reduing the delay as a timer with evry iteration of the loop

delay1-=1

if(delay1==0): #Delay 0 states that a process is complete

#So, as and when the process is done the client sends a message to the server.

print(client[0].recv(2000).decode('ascii'))

order+=1 #Keeps counting the no. of orders delivered to their locations.

delay2-=1

if(delay2==0):

print(client[1].recv(2001).decode('ascii'))

order+=1

delay3-=1

if(delay3==0):

print(client[2].recv(2002).decode('ascii'))

order+=1

if(delay1<=0 and delay2<=0 and delay3<=0 and order==n):

break

# Closing all the client connections

client[0].close()

client[1].close()

client[2].close()

f.close()

def main():

# function to accept all the connections from clients

accept\_connections()

# takes input of delivery location

take\_input()

# write assigned task information to output file

write\_output()

# assign task to trucks based on availability

assigning\_task()

main()

**CODE Images:**





