**UNIVERSITETI I PRISHTINËS**

**Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike**

**Departamenti i Kompjuterikës**



**Projekti 1**

**Lënda: Rrjetat kompjuterike**

**Profesor:** Blerim Rexha **Studenti: Veton Shala**  **Asistent:** Haxhi Lajqi **Nr.Index:160714100082**

**Prishtinë, Prill 2018**

**HYRJE**

Fokusi i këtij projekti është në të kuptuarit e mënyrës se si bëhet komunikimi në mes serverit dhe klientit permës sockets duke marrë për bazë TCP-n dhe UDP-n . E tërë kjo është bërë me veglën Visual Studio në gjuhën e lartë programuese Python andaj I tërë problem I sigurimit të lidhjes dhe mënyrës se si pergjigjja vie deri tek klienti është shtruar në të lartëcekurat , duke pasur parasysh të gjitha rregullat dhe konditat e parashtruara në projekt.

Pasi është siguruar lidhja në mes klientit dhe serverit për kushtet që janë paraparë në projekt , është bërë zgjidhja e atillë që varësisht se cfarë operacioni deshiron të kryej klienti të zgjedh numrin e tij dhe në baze të kësaj, kërkesa të dergohet në server , ku aty më pastaj janë kryer llogaritjet dhe është siguruar përgjigjja e cila përsëri është kthyer tek klienti .

Përfitimet nga ky projekt kanë qenë mjaftë të mëdha për arsye së ne cdo ditë jemi të rrethuara nga këto veprime , lidhje dhe komunikime, mirëpo në mënyrën se si janë arritur këto nuk kemi pasur njohuri sa kemi tani , andaj edhe pse nuk ishte ndonjë projekt I gjerë ishte shumë fitimprurëse për dijën tone .

**PERSHKRIMI**

**Socket Programming** paraqet një mënyrë për të lidhur dy nyje (nodes) në një rrjet për të komunikuar me njëri tjetrin.Ky programim fillon duke importuar librarian socket dhe duke bërë një socket të thjeshtë.

import socket

Njohuritë lidhur me sockets dhe protokolet ndihmojnë në një masë të madhe në realizimin e projektit. Shumë herë është përmendur që socket paraqesin një llojë porte në rrjet , andaj rëndësia e tyre është mjaftë e madhe. Nje socket ka numrin e portit(Port Number) dhe Ip adresen (IP Address) . Mund të përdorën protokolet TCP dhe UDP. E para ofron një kanal të komunikimit të sigurt , ndërsa ajo e dyta nuk ofron siguri në këtë kanal. TCP ka shumë arsye pse duhet të përdoret , mirëpo UDP ofron një shpejtësi më të madhe.

I tërë projekti në përgjethësi ështe i bazuar ne mbushjen e një array me tw dhëna dhe perpunimin e tij, si dhe me try dhe while.

**TCP-KLIENTI**

E dimë që Ip adresa e identifikon një host mirëpo për të lejuar komunikimin nuk mjafton vetëm kjo për arsye se Sistemi Operativ pyet se kush duhet ta marre atë mesazh, pra duhet te dihet sakt edhe procesi , duke e ditur Port Number ose Socket Number.

Në bazë të kërkesës në projekt është përdorur “localhost” si host server dhe porti 12000 si port I nënkuptushëm.

Në pjesën e kodit fillimisht janë përfshirë

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

AF\_\* dhe SOCK\_\* paraqesin familjen(AddressFamily) dhe tipin e Socketit(SocketKind) .

AF\_INET paraqet opcionin më të sigurtë në internet dhe I referohet adresës së familjës ipv4. SOCK\_STREAM na lë të kuptohet që është një TCP socket.

Përmes input() është bërë e mundshme që klienti te ndryshoj emrin e serverit dhe portin sipas nevojës.

Shfrytëzuesi është në gjendje të thërrash kërkesën IPADRESA ,NUMRIIPORTIT, ,BASHTINGLLORE , PRINTIMI ,EMRIIKOMPJUTERIT, KOHA , LOJA , FIBONACCI , KOVERTIMI,MOSHAEKLIENTIT, KTHEFJALIN.

Shfytëzuesi është në gjendje të përfundoj punën përmes klikimit te numrit 0

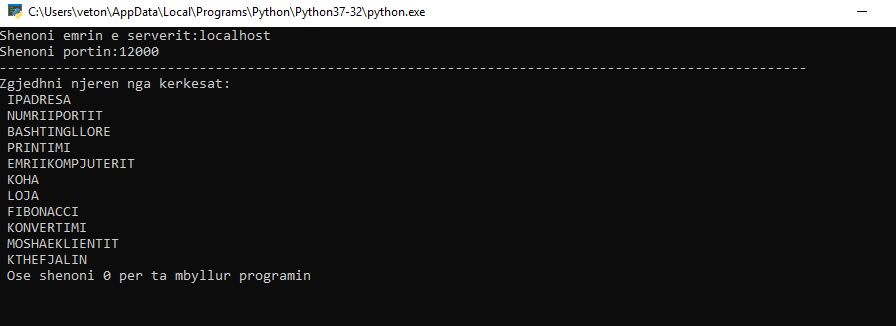
sendall() – shfytëzohet për të dërguar të dhënat nga klienti në server dhe anasjelltas në rastin kur është TCP

Në secilin rast pranohet një mesazh jo më I gjatë se 128 karaktere(bajta,1024bita) , Klientit I është ofruar mundësia që në bazë te operacioni që ai dëshiron ta kryej të shtyp emrin e asaj kerkese dhe ajo kërkesë e tij të dërgohet te klienti nga serveri,

Po ashtu është cekur që nëse server nuk mund ta pranoj atë kerkesë në fund është perfshirë edhe kjo permes print(""Kerkesa nuk mund te jete me e gjate se 128 karaktere!"),

Ose ne raste kur nuk shtypet ndonje kerkes ateher është perfshirë shtypja ne ekran

print("Ju lutem shenoni nje kerkese!");



**TCP SERVERI**

Në pjesën e kodit fillimisht janë importuar disa librari dhe ato jane:

import socket

import sys

from \_thread import \*

from datetime import datetime

import random

import math

Për të siguruar kërkesën e cekur në projekt që të sigurohet lidhja dhe komunikimi jo vetëm në një paisje dhe për të përdorur esencën kryesore të sockets është përfshirë libraria import\_thread . Libraria import time për funksionin KOHA në mënyrë që të realizohet kërkesa që të merret koha aktuale e serverit, import random në rastin LOJA pra është importuar libraria random duke zgjedhur nje numer te cfardoshum , import datetime në mënyrë që të merret edhe data , kjo është shfytezuar tek rasti KOHA,import math eshte shfrytezuar per funksionin e llogaritjes se moshes se klientit,

Host=’’, serverPort=12000,

Hosti nuk eshte i caktuar pasi qe kemi perodrur \_thread dhe ne te njejten kohe mund te jene shume host te cilet e perdorin serverin, nese e caktojm hostin athere serverin mund ta perdore vetem ai i cili eshte i caktuar ne Host.

try :

serverSocket.bind((host,serverPort)).: është një metodë që lidh adresën me socket.

Exept socket.error as err:print(str(err)) :Eshte perdorur ne rast te ndonje errori te lidhjes me socket.

.accept() :pranon “pasivisht” lidhjen e klientit TCP, duke pritur derisa lidhja te arrin .

serverSocket.listen(10); Serveri vendoset ne formen e ndegjimit te kerkeses nga klienti -TCP-LISTENER

Një funksion në Python deklarohet me def , andaj në server të gjitha kërkesat që janë perfshirë në projekt janë bërë me funksione duke I emertuar ato dhe varesisht nga rasti , janë marrë me parametra apo pa parametra.

def Klient\_Thread(kerkesa,conn, addr):

paraqet funksionin i cili ka tre parametra dhe në vete ka të gjitha funksionet e kërkuara .Gjithashtu thread bene te mundur perdorimin e serverit nga 2 e me shume klienta ne te njeten kohe.

Më poshtë shihen shqyrtimi i funksioneve.

METODA IPADRESA

if(kerkesa[0]=='IPADRESA'):

conn.send(str.encode(" IP Adresa e klientit është:"+addr[0]));

IP Address-a e klientit është marr përmes addr[0] për arsye se sic shihet edhe nga më lartë merret Ip-ja e klientit , pastaj kjo dergohet permes conn.send() , pjesa brenda kllapav paraqet enkodim dhe str() paraqet një nga shumë metodat e stringjev në python , encode() metoda kthen një version të enkoduar të strigut, Pythom by default encodimin e ka ASCII.

METODA NUMRIIPORITIT

elif(kerkesa[0]=='NUMRIIPORTIT'):

conn.send(str.encode("Klienti është duke përdorur portin "+str(addr[1])));

Numri i portit te klietit mirret nga addr[1] .Pastaj me ane te conn.send() dergohet tek klienti.

METODA BASHTINGLLORET

elif(kerkesa[0]=='BASHTINGLLORE'):

try:

inputi="";

inputi=inputi.join(kerkesa[1]); #ruajme fjaline ne nje string

count = 0;

bashtinglloret = set("bcdfghjklmnpqrstvxzBCDFGHJKMNPQRSTVXZ");

for letter in inputi: #iterojme neper cdo shkronje te stringut

if letter in bashtinglloret: #nese shkronja ne iterim gjindet tek bashtinglloret rritet count

count += 1;

conn.send(str.encode("Teksti i pranuar përmban "+ str(count) +" bashtingllore"));

except IndexError:

conn.send(str.encode("Shenoni nje fjali pas kerkeses

BASHTINGLLORE!"))

Më së pari pranohet teksti me anë te input=””, pastaj e ruajm këtë tekst në një string ,

e deklarojmë një “numerues” count=0, po ashtu deklarojme atributin bashtinglloret te cilit i japim vlerat e bashtinglloreve, dhe për qdo shkronje në tekst që është e njejtë me bashtinglloret counti do të rritet për një , ne fund e printojmë numrin e bashtinglloreve.

METODA PRINTIMI

elif(kerkesa[0]=='PRINTIMI'):

try:

inputi="";

inputi=str.join(" " , kerkesa[1]);

conn.send(str.encode("Fjalia e dhene per printim "+inputi));

except IndexError:

conn.send(str.encode("Ju lutem shenoni nje fjali pas kerkeses PRINTO"));

Metoda PRINTIMI nuk ka parametra dhe është një metode e thjeshtë tek e cila pranohet teksti i dhënë nga shfrytëzuesi dhe pastaj i njejti dërgohet tek klienti.

METODA EMRIIKOMPJUTERIT

elif(kerkesa[0]=='EMRIIKOMPJUTERIT'):

try:

emriikompjuterit=socket.gethostname();

conn.send( str.encode("Emri i kompjuterit është "+emriikompjuterit));

except error:

conn.send(str.encode("Emri i kompjutert nuk dihet."))

Funksioni gethostname() kthen një string që përmban emrin e hostit të makinës ku interpreteri i Pythonit është duke u ekzekutuar. Keshtu dergohet pastaj tek klienti.

METODA KOHA

elif(kerkesa[0]=='KOHA'):

koha=datetime.now().strftime('%Y-%m-%d %H:%M:%S');

conn.send(str.encode(koha));

Koha reale është marrë përmes funksioneve të gatshme që I ofron python , duke pasur parasysh librarian e importuar arrihet që të kemi mundësine e shfytëzimit të tyre dhe pastaj sic shihet kjo kohë dërgohet tek shfytëzuesi,

METODA LOJA

elif(kerkesa[0]=='LOJA'):

srand= '';

for x in range(7):

rand= random.randint(1,49); #numer i zakonshem

randString = str(rand) + " "; #konvertohet ne string

srand += randString; #te gjithe numrat ne string

conn.send(str.encode(srand));

Me kerkesa[o] deklarohet një array e zbrazët në python , ndërsa me srand kemi shtuar elementere të reja në listë , përmes libraris import random janë siguruar 7 numra të rastit nga 49 gjithesejt dhe janë dërguar tek klienti permes conn.send().

METODA FIBONNACI

Më se pari është definuar funksioni FIBONNACI me parametër numri, me anë te kushteve if else ështe berë shqyrtimi se a është numri më i vogel se barazë me 1, nese po athere e kthen numrin nese jo atherë return bënë nje llogaritje duke ja hequr numrit 1 në vleren e parë dhe 2 në vleren e dytë dhr duke i mbledhur ato, pastaj mbreda funskionit behet inputi i numrit nga jasht dhe pritntimi i tij ne ekran

def FIBONNACI(numri):

if numri <= 1:

return numri;

else:

return(FIBONNACI(numri-1) + FIBONNACI(numri-2));

\*Mbrenda funskioni perpunimi\_kerkeses

elif(kerkesa[0]=='FIBONACCI'):

try:

numri=FIBONACCI(int(kerkesa[1]));

conn.send(str.encode(str(numri)));

except IndexError:

conn.send(str.encode("Ju lutem shenoni nje shifer pas kerkeses FIBONACCI"));

except ValueError:

conn.send(str.encode("Ju lutem shenoni nje shifer pas kerkeses FIBONACCI"));

METODA E KONVERTIMIT

def KONVERTIMI(inputi,outputi):

if(inputi=="Klioëatt-to-Hoursepower"):

return outputi\*1.34;

elif(inputi=="Horsepower-to-Kilowatt"):

return (outputi/1.34);

elif(inputi=="Degrees-to-Radian"):

return (outputi\*1.0174);

elif(inputi=="Radian-to-Degrees"):

return outputi/1.0174;

elif(inputi=="Gallons-to-Liters"):

return (outputi\*3.785);

elif(inputi=="Liters-to-Gallons"):

return (outputi/3.785);

\*Mbrenda funskionit perpunimi\_kerkeses

elif(kerkesa[0]=='KONVERTIMI'):

Stringu="Mundesite per konvertime:\nKliowatt-to-Hoursepower \nHorsepower-to-Kilowatt \nDegrees-to-Radian \nRadian-to-Degrees \nGallons-to-Liters \nLiters-to-Gallons";

try:

input=kerkesa[1];

outputi=float(kerkesa[2]);

conn.send(str.encode(str(KONVERTIMI(input,outputi))));

except IndexError:

conn.send(str.encode("Ju lutem shenoni cka deshironi te konvertoni pastaj shifren! \n"+Stringu));

except ValueError:

conn.send(str.encode("Ju lutem shenoni cka deshironi te konvertoni pastaj shifren! \n" +Stringu));

Klientit fillimisht i kërkohet që të shtyp emrin e metodes e pastaj llojin e konvertimit që ai dëshiron ta bëjë dhe vleren te cilen deshiron ta konvertoj, pastaj në funksionin Konverto(funksion me dy parametra,inputi-hyres dhe outputi-dales) varësisht nga lloji i konvertimit llogaritet dhe arrihet rezultati i deshiruar .

METDA MOSHAEKLIENTIT

elif(kerkesa[0]=='MOSHAEKLIENTIT'):

try:

vitiilindjes= int(kerkesa[1]);

vitimomental =2019;

mosha = (vitimomental-vitiilindjes);

conn.send(str.encode(str(“Mosha e klienti eshte “ +mosha)));

except IndexError:

conn.send(str.encode("Shenoni Vitin e lindjes pas kerkeses MOSHAEKLIENTIT"));

except ValueError:

conn.send(str.encode("Ju lutem shenoni njeren nga kerkesat"));

Klientit me se pari i kerkohet te shenojn moshen e ti pastaj kemi dekalruar vitin momental si 2019, me anw të një funskioni zbret vitin momental me vitin e lindjes dhe jep moshen e klineiti

METODA KTHEFJALIN

elif(kerkesa[0]=='KTHEFJALINE'):

try:

fjalia="";

fjalia=str.join(" " , kerkesa[1]);

fjalia=str.reverse();

conn.send(str.encode("Fjalia e dhene per ta kthyer eshte "+fjalia));

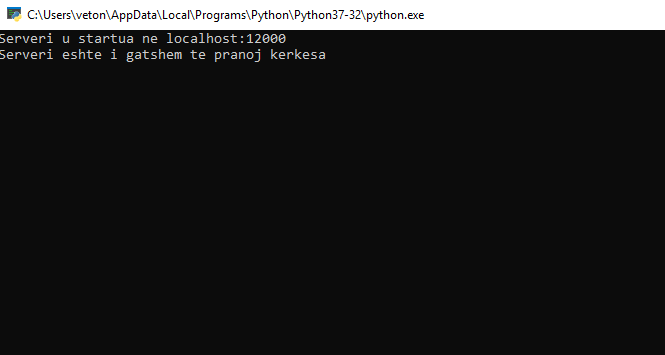
except IndexError:

conn.send(str.encode("Ju lutem shenoni nje fjali pas kerkeses KTHEFJALIN"));

except ValueError:

conn.send(str.encode("Ju lutem shenoni njeren nga kerkesat"));

Në filim klienti jep një fjali ne array , pastaj me anë të metodes reveres ajo fjali kthehet prapa dhe printohet.

Në fund kemii perdorur While true , ku perderisa konektimi ne mes serverit dhe klientit është në rregull athere në ekran do te shtypet si ne fig me poshte

**UDP KLIENTI**

Pasiqë UDP nuk pranon shumë kërkesa por vetëm një kërkesë për një herë , në pjesën e kodit janë bërë disa ndryshime .

Fillimisht sikurse edhe tek TCP klienti janë perfshirë libraritë e nevojshme

import socket

Te cilat u shtjelluan më në detaje edhe në pjesën me lartë.

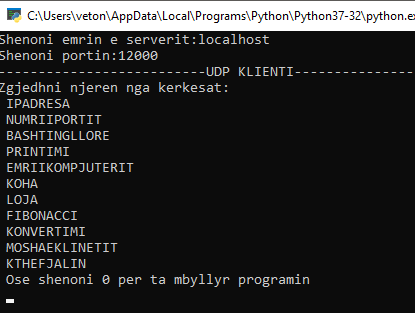
s=socket(AF\_INET,SOCK\_DGRAM)

SOCK\_DGRAM nënkupton që është një UDP socket .

Tek pjesa e klientit kodi është i thjeshtë me print() janë dhënë disa sqarime së mënyrën se si duhet të trajtohet kërkesa , dhe pastaj permes var=input("Zgjedhni njeren nga kerkesat”)është bërë e mundur që shfytëzuesi ta shkruaj emrine metodes qe deshrion ta perodri.

Të dhënat që pranohen përmes recv , ashtu sic është përshkruar në kërkesë. Pastaj decode-hen në mënyrat e shprehura më larte , dhe të cilat janë cekur edhe në TCP.

s.close() – mbylle socketin

Pamja e parë e UDP Klientit shihet më poshtë

**UDP SERVERI**

Në fillim janë importuar librarit e nevojshme

from socket import \*

import random

import socket

from datetime import datetime

import random

import time

import re

import string

import \_thread

import math

iimport re- importon librarin e Regular Expressions

import string- importon librarin string , ndersa te tjerat janë përshkruar edhe tek TCP Serveri.

Funksionet e para pra për të siguruar IP adresen ,Numri i poritit dhe Numrin e bashtinglloreve janë njesoj sikurse tek TCP Serveri.

METODA PRINTIMI nuk mund të trajtohet në mënyrë të thjeshtë sic është trajtuar në TCP andaj është shprehur ndryshe.

elif(kerkesa[0]=='PRINTIMI'):

try:

kerkesa[1] = kerkesa[1].strip();

server\_socket.sendto(str.encode("Fjalia e dhene per printim "+kerkesa[1]), addr);

except IndexError:

server\_socket.sendto(str.encode("Ju lutem shenoni nje fjali pas kerkeses PRINTIMI!"), addr);

Funksionet LOJA ,EMRIIKOMPJUTERIT dhe KOHA janë shqytuar po ashtu në mënyrë të njejtë sikurse tek TCP-Serveri .

METODA FIBONACCI

elif(kerkesa[0]=='FIBONACCI'):

try:

print(int(kerkesa[1]));

numri=FIBONACCI(int(kerkesa[1]));

server\_socket.sendto(str.encode(str(numri)), addr);

except IndexError:

server\_socket.sendto(str.encode("Shenoni nje numer pas kerkeses FIBONACCI"), addr);

except ValueError:

server\_socket.sendto(str.encode("Shenoni nje numer pas kerkeses FIBONACCI"), addr);

METODA KONVERITIMI

Pjesa e parë e metodes konvertimi është e njejt me atë te TCP-SERVERI ndersa pjesa e dytë ndryshon

elif(kerkesa[0]=='KONVERTIMI'):

string="Mundesite per konvertime:\nKliowatt-to-Hoursepower \nHorsepower-to-Kilowatt \nDegrees-to-Radian \nRadian-to-Degrees \nGallons-to-Liters \nLiters-to-Gallons";

try:

inputi=kerkesa[1];

outputi=float(kerkesa[2]);

server\_socket.sendto(str.encode(str(KONVERTIMI(inputi,outputi))), addr);

except IndexError:

server\_socket.sendto(str.encode("Ju lutem shenoni cka deshironi te konvertoni pastaj shifren! \n" +string), addr);

except ValueError:

server\_socket.sendto(str.encode("Ju lutem shenoni cka deshironi te konvertoni pastaj shifren!\n "+string), addr);

METODA MOSHAEKLIENTIT

elif(kerkesa[0]=='MOSHAEKLIENTIT'):

try:

vitiilindjes= int(kerkesa[1]);

vitimomental =2019;

mosha = (vitimomental-vitiilindjes);

conn.send(str.encode(str("Mosha e klienti eshte"+mosha),addr));

except IndexError:

conn.send(str.encode("Shenoni Vitin e lindjes pas kerkeses MOSHAEKLIENTIT"),addr);

except ValueError:

conn.send(str.encode("Ju lutem shenoni njeren nga kerkesat"),addr);

METODA KTHEFJALINE

Pra kjo metode sikurese te TCP-serveri pranon nje tekst dhe ate tekst e kthen mbrapa , eshte shtuar add rose ndryshe addresa

elif(kerkesa[0]=='KTHEFJALINE'):

try:

fjalia="";

fjalia=str.join(" " , kerkesa[1]);

fjalia=str.reverse();

conn.send(str.encode("Fjalia e dhene per ta kthyer eshte "+fjalia),addr);

except IndexError:

conn.send(str.encode("Ju lutem shenoni nje fjali pas kerkeses KTHEFJALIN"),addr);

except ValueError:

conn.send(str.encode("Ju lutem shenoni njeren nga kerkesat"),addr);

**PERFUNDIMET**

Nga ato që pame , u arritën shumë përfundime lidhur me projketin dhe TCP e UDP sockets , punimi I këtij projekti na ofroj të njihemi më shumë me “parapvijat “ që ndodhin pas cdo përdoruesi që përdor internetin . Gjatë shqyrtimit të projektit përkatesisht definimit të funksionev është marrë vetëm nga një zgjidhje e mundshme për secilin nga funksionet , ofrohen edhe shumë zgjidhje të tjera mirepo normalisht është marrë ajo që për ne si student duket më e përshtatshme .Përvec të tjerash u mësuan përparësite që kanë dy protokolet. TCP ofron Reliable Data Transfer Services- proceset e komunikimit mund të mbështetën në TCP për të ofruar të gjitha të dhënat e dërguara pa gabime dhe nën renditjen e duhur . UDP siguron shërbime minimale , e cila është edhe connectionless që do të thotë që nuk ka handshaking përpara se proceset të komunikojnë , siguron një kanal të komunikimit jo të besueshëm , që do të thotë që kur dërgohet një mesazh në socketin e UDP-së , UDP nuk sihuron asnjë garanci që mesazhi do te arrijë ne procesin e pranimit , gjëra që TCP-ja i bënë .Pra UDP nuk siguron connection setup , reliability, flow control, congestion control, timing , throughput garantee, security. E vetmja përparësi e tij është që është më e shpejtë.

**BURIMET E INFORMACIONIT**

N kete projekt jane shfrytuar burime te shumta per qellime te ndryshme te plotesimit te detyre qe ka projekti.

Computer Networks and Internets by Douglas E.COMER

Computer Networking a Top-Down Approach by Kursore and Ross

<https://www.pythoncentral.io/>

<https://www.geeksforgeeks.org/python-programming-language/>

<https://www.codecademy.com/en/tracks/python>

<https://www.python.org/doc/>

<https://www.youtube.com/watch?v=F-_PKUUM-qY>

<https://docs.python.org/3/library/re.html>

<https://docs.python.org/3/howto/regex.html>

<https://www.tutorialspoint.com/python/python_reg_expressions.htm>

<https://www.khanacademy.org/computing/computer-programming>