UNIVERSITETI I PRISHTINES

FAKULTETI I INXHINIERISË ELEKTRIKE DHE KOMPJUTERIKE

SOCKET PROGRAMMING

Client-Server Design

LËNDA RRJETA KOMPJUTERIKE

Student: Blina Smailaj Profesor: Dr.techn.Blerim Rexha

ID :190714100183 Asistent: Msc. Haxhi Lajqi

Prishtine

2 maj 2021

ABSTRACT

Ky projekt ka për qëllim realizimin e protokolleve *FIEK TCP* dhe *FIEK UDP* , përmes të cilave mundësohet lidhja server-klient , duke përdorur gjuhen programuese Python dhe libraritë e gatshme të kësaj gjuhe , perkatesisht soketat dhe funksionet e gatshme të saj.

Me anë të serverit mundësohet komunikimi me klientë të ndryshëm .Serveri punon pa ndërprerje, ndërsa klienti gjithmonë mund të shkëputet apo të ri-lidhet me serverin , poashtu duke I mundësuar klientit të caktojë vetë portin dhe IP-Adresen edhe pse serveri të cilin kemi implementuar gjendet në local host me portin 14000 . Mirëpo nëse do të kishim serverë të tjerë kjo mundësi do të ishte shumë e përdorshme për klientin .

Sapo klienti dhe serveri të lidhen në socketat përkatës , serveri qëndron në pritje te pafundme për kërkesa te klientit , deri sa ai vetë të mbyllë dhe lidhjen me socketin e serverit .

Serveri është gjithmonë I gatshëm për të kthyer përgjigje , si dhe nëse kerkesat nuk jane valide , ai dërgon një mesazh informues duke e orientuar drejt dhënies së saktë të kërkesës së tyre .

Metodat e implementuara në server dhe të qasshme për shfrytëzuesin janë *IP,NRPORTIT,NUMERO,ANASJELLTAS,PALINDROM,KOHA,LOJA,GCF,KONVERTO ,HASHPWD* dhe *PLATFORM*

Përmbajtja

[**Dizajnimi Klient-Server**  1](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426259)

[Abstrakt 2](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426260)

[Permbajtja: 3](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426261)

[Hyrje: 4](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426262)

[FIEK PROTOKOLLI 5](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426263)

[FIEK- TCP 5](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426264)

[FIEK-UDP 10](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426265)

[Metodat: 12](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426266)

[1.IP 12](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426267)

[2.NRPORTIT 12](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426268)

[3.NUMERO 12](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426269)

[4.ANASJELLTAS 13](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426270)

[5.GCF 14](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426271)

[6.PALINDROM 14](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426272)

[7.KOHA 15](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426273)

[8.LOJA 15](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426274)

[9.KONVERTO 15](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426275)

[10.HASHPWD 16](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426276)

[11.PLATFORM 17](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426277)

[Testimi i metodave 18](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426278)

[REFERENCAT 20](file:///C:\Users\User\Downloads\Kushtrim_Haziri_Raporti.docx#_Toc37426280)

HYRJE

**TCP (Transmission Data Protocol ) –** Lidhja TCP stabilizohet me anë të ‘three way handshake ‘ . Është proces i inicializimit dhe ‘acknowledgmentit’ të lidhjes . Sapo të krijohet handshake fillon transferimi I të dhënave dhe sapo transmisioni të perfundojë , lidhja ndërpritet duke mbyllur një qark virtual .

**UDP (User Datagram Protocol ) –** UDP përdor një metodë të thjeshtë transmisioni pa handshakes , reabilitet dhe pa integritet të të dhënave . UDP gjithashtu supozon që kontrolli dhe korrigjimi i gabimit nuk është i rëndësishëm ose nuk kryhet në aplikim, për të shmangur shpenzimet e përgjithshme të një përpunimi të tillë në nivelin e ndërfaqes së rrjetit.

TCP: Ju ndihmon të krijoni / vendosni një lidhje midis llojeve të ndryshme të kompjuterëve.

Operon në mënyrë të pavarur nga sistemi operativ

Mbështet shumë protokolle rutimi.

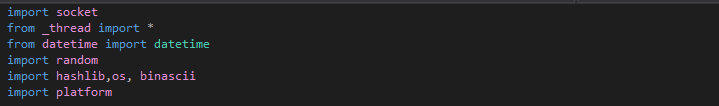
Mundëson punën në internet ndërmjet organizatave.

UDP: Metoda UDP përdoret kryesisht nga aplikacione ‘time-sensitive’, si dhe nga servera që u përgjigjen pyetjeve të vogla nga një bazë më e madhe klientësh.

UDP është i pajtueshëm me transmetimet e paketave për dërgimin në të gjithë rrjetin dhe për dërgimin me shumë kast.

Përdoret gjithashtu në Domain Name System, Voice over IP dhe online games.

Modeli *klient-server* është një strukturë e shpërndarë e aplikacioneve që ndan detyrat ose ngarkesat e punës midis ofruesve të një burimi ose shërbimi, të quajtur servera dhe kërkuesve të shërbimeve, të quajtur klientë. Një klient zakonisht nuk ndan asnjë nga burimet e tij, por ai kërkon përmbajtje ose shërbim nga serveri. Prandaj, klientët fillojnë sesione komunikimi me servera, të cilët presin kërkesat e ardhshme. Ky model eshte implementuar ne kete projekt dhe do te shohim se si punon ne faqet ne vijim.

FIEK PROTOKOLLI

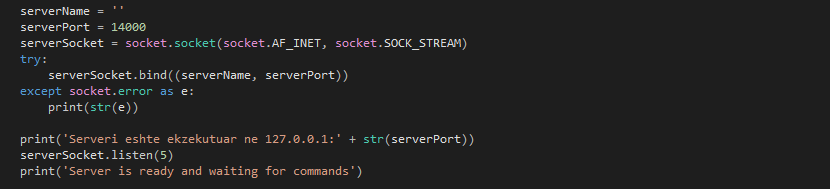
Për zhvillimin e protokolit te FIEK-ut janë përdorur libraritë e gatshme si në vijim .

Libraria socket është implementuar për krijimin e socketave dhe për të shfrytëzuar metodat e gatshme të saj si socket.AF\_INET ,socket.SOCK\_STREAM , socket.SOCK\_DGRAM etc .

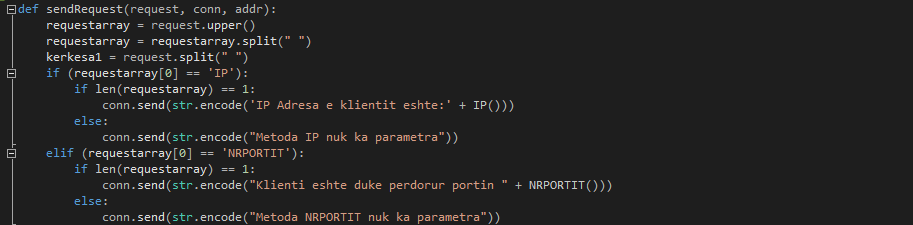
Libraria \_thread është implementuar për të mundësuar multithreading ( komunikimin e njëkohshëm me klienta të ndryshëm ). Ndërsa libraritë e tjera janë implemetuar për metodat e serverit ( lib random është shfrytëzuar tek loja – hashlib,loja,binascii tek hashpwd – platform tek platform)

FIEK-TCP

TCP (Transmision Control Protocol) është një standard që përcakton se si vendoset dhe mirëmbahet një bisedë në rrjetë përmes edhe së cilës programet mund të dërgojne të dhëna .

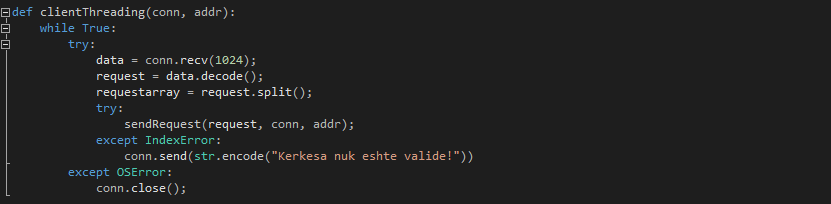
FIEK-TCP-Server

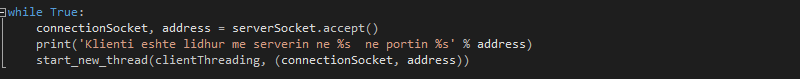
Fillimisht inicializohen në variabla serveri dhe porti në të cilin do të komunikoje ai me klientin . AF\_INET paraqet Adress Family Internet I cili tregon se socketi do të komunikojë me anë të IP adresave Ipv4 ndërsa SOCK\_STREAM tregon se komunikimi bëhet përmes TCP (connection based protocol). Metoda bind() e inicializon socketin me IP adresën dhe portin përkatës .Kjo metodë përdoret për socketin e Serverit. Nëse gjate funksionimit të këtij kodi eventualisht ka ndonjë gabim përmes blloqeve try ,except tregojmë se pati ndonjë problem gjatë kodit.

Metoda listen() tregon se sa është rradha e klientëve të cilët mund të marrin përgjigje nga serveri , në këtë rast serveri mund të dëgjojë deri në 5 klientë .

Funksioni sendRequest ka për detyrë të dërgojë pergjigje për kërkesën e klientit . Parametri request është kërkesa hyrëse nga klienti . Parametrat conn dhe addr jane vlerat kthyese te metodes accept() , ku conn paraqet një socket object që përdoret përkohesisht për dërgimin e të dhënave në lidhje , ndërsa addr është adresa e lidhur per socketin në anën tjetër të lidhjes .

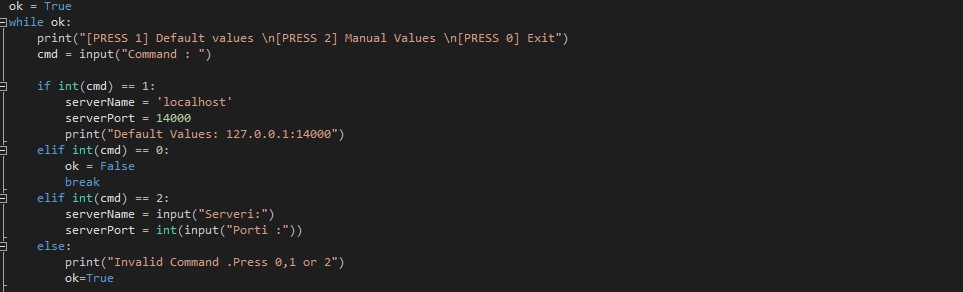
Variabla requestarray fillimisht konvertohet në shkronja të mëdha dhe përmes metodes split ndan fjalët të cilat na shërbejnë për pjesë të mëtejshme te kodit . Le të shqyrtojmë vetëm rrjedhën e metodës së parë , pasiqë edhe metodat e tjera rrjedhin ngjajshmërisht .

Nëse fjala e parë eshte ‘IP’ vazhdojmë me këtë metodë , përndryshe serveri kalon në kushtëzimet e tjera . Nëse gjatësia e kërkesës është e barabartë me një (që do të thotë se user-I ka shtypur vetëm : IP) , atëherë dërgojmë kërkesën e enkoduar në socketin e përkohshem conn përmes metodes send (kjo metode perdoret per dergimin e informatave nga nje socket ne tjetrin ) si dhe concatenojme përgjigjen të cilën na e kthen metoda IP() për të cilen do të flasim dhe më vone . Përndryshe nëse gjatësia e kërkesës nuk është e barabartë me një , klientit i dërgojme mesazhin validues dhe kërkojmë të japë përsëri kërkesën .

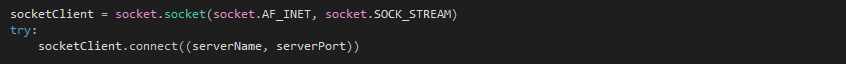
Qëllimi i kësaj metode është që të krijojmë një mundësi që serveri të mund të kontaktojë me disa klientë njëkohësisht dhe pa i ngatërruar kërkesat . Metoda recv shërben si ‘buffer’ duke e ndarë kërkesën në 1024 byte buffer sizes . Të dhënat dekodohen dhe dërgohen tek metoda sendRequest për tu shqytuar dhe me tej për tu dërguar tek klienti . Përndryshe nëse paraqitet Index Error dergohet mesazhi perkates , ndërsa në rast te OSError socketi I klientit mbyllet.

Metoda accept përdoret për të treguar se është pranuar kërkesa e cila po I drejtohet serverit . Kjo metodë kthen dy argumente të cilat u sqaruan në kodin më lartë . Pasiqë pranohet kërkesa shtypim një mesazh me IP adresën e serverit dhe portin e lidhjes .

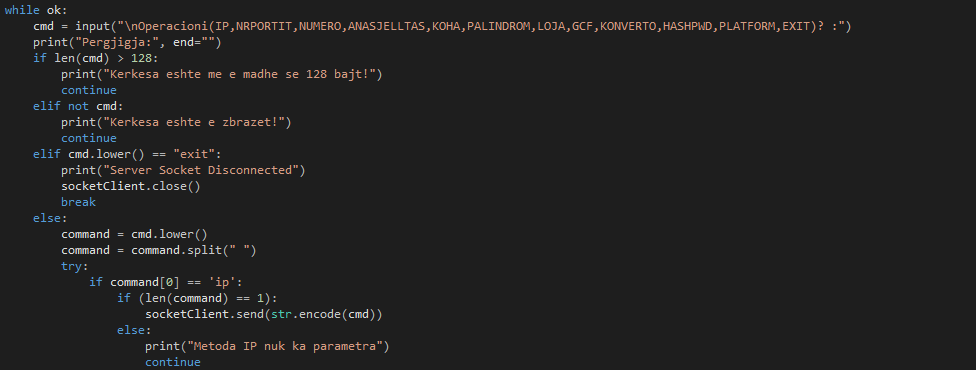
Metoda start\_neë\_thread krijon një thread te ri dhe ekzekuton funsionin clientThreading me parametrat connectionSocket dhe address .

FIEK-TCP-Client

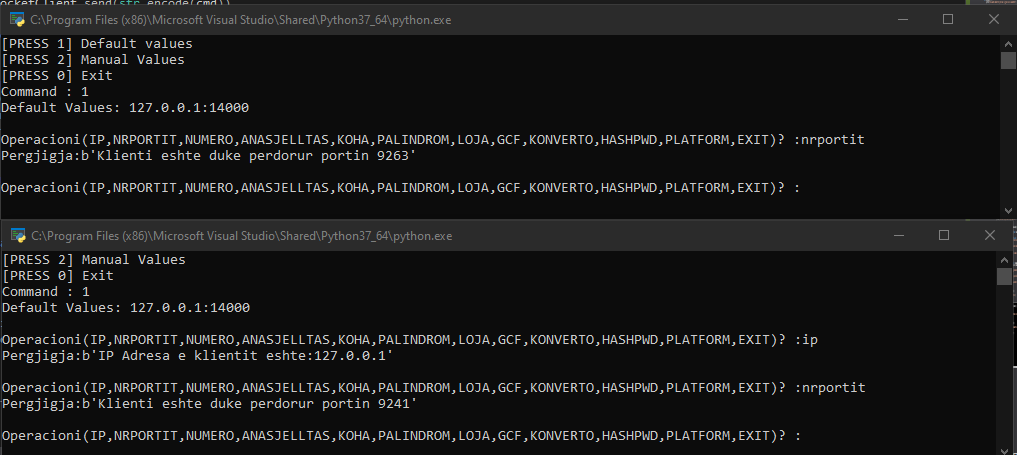
Në ketë pjesë të kodit klienti fillimisht duhet të japë një informatë e cila vendos se cili është porti dhe serveri me të cilin do të lidhet qofshin këto manuale apo default , si dhe opsionin exit per ta mbyllur ekzekutimin . Ne qoftë se useri nuk jep nje nga mundesite 0 , 1 apo 2 ok = true dhe unaza përseritet .

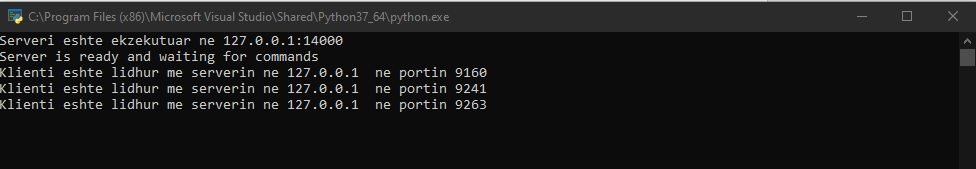


Ngjashëm si tek serveri , klienti krijon socketin me familjen e adresave Ipv4 dhe TCP connection . Metoda connect() krijon nje 3 ëay handshake me serverin e socketit dhe krijon një lidhje .

Perndryshe nëse nuk zbatohet kjo metodë , lidhja nuk stabilizohet dhe programi kthehet përsëri në fillim tek incializimi I portit dhe serverit .

Më pas në variablën cmd ruhet kërkesa të cilen e jep klienti . Varësisht nga hyrja kjo variabël kalon në disa validime . Nëse gjatësia është me e madhe se 128 byte printohet mesazhi dhe programi kthehet përsëri tek futja e kerkesës se re . Në qoftë se komanda është ‘exit’ socketi I clientit mbyllet dhe unaza ndalet . Me pas në variablën command ruhet kërkesa me shkronja te vogla si dhe ndahet ne fjalë të cilat më pas mund ti shfrytezojmë si array . Në qoftë se fjala e parë është ‘ip’ dhe gjatësia e komandës është nje , klienti dërgon kërkesën e enkoduar tek serveri për të marrë pergjigjen , përndryshe vazhdon në kushtëzimet e tjera me validimet përkatëse .

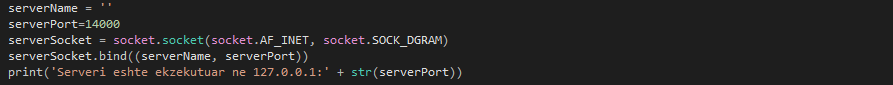
Komunikimi me dy klientë :

Serveri :

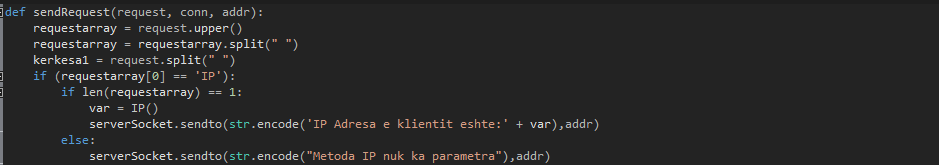
FIEK-UDP

UDP (User Datagram Protocol) është një protokoll komunikimi që përdoret kryesisht për krijimin e lidhjeve te shpejta por toleruese ndaj humbjeve te paketave ndermjet aplikacioneve në internet.

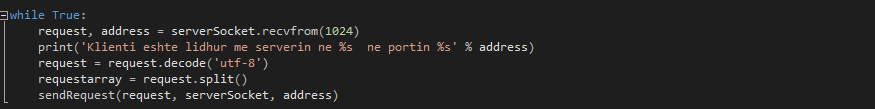
FIEK-UDP-Server



Njëlloj si tek FIEK-TCP-Serveri fillimisht janë inicializuar variablat serverName dhe serverPort përkatësisht IP adresa e serverit dhe porti në të cilin do të komunikojnë . Variabla serverSocket është socket I familjes (Adress Family Internet) se Ipv4 ndërsa SOCK\_DGRAM tregon se socketi do të komunikojë përmes UDP(connectionless protocol) .



Dallimi nga metoda sendRequest tek TCP është metoda sendto() e cila përdoret ne socketa ‘connectionless’ pra për të dërguar Datagrams në UDP socket . Kjo metodë mund të përdoret për të dërguar të dhëna nga të dyja anët (klient-server) , ndërsa variabla addr përmban IP adresën dhe portin e socketit . Pjesa e metodave është pothuajse e njejtë me ato te shpjeguara më larte prandaj nuk do ti shqyrtojmë në këtë pjesë .



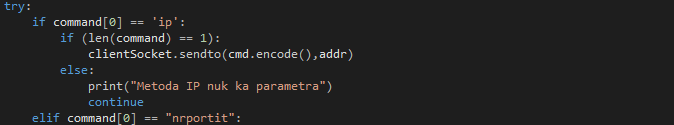
Në këtë pjesë Serveri pret për kërkesa të reja . Metoda recvfrom(1024) tregon se sa bajt do te lexojë UDP socketi . Në variablen request do te ruhet bytes objekti i lexuar nga UDP socketi ndërsa në variablen address ruhet adresa e socketit të klientit . Kerkesa dekodohet në ‘utf-8’ per shmangien e gabimeve , ndahet me metoden split dhe dergohet në metoden SendRequest . Metoda sendRequest do të thirret sa herë qe serveri do te jete I gatshem te pranoje pergjigje .

FIEK-UDP-Client

Në këtë pjesë do të shqyrtojmë vetëm ndryshimet nga FIEK-TCP-Client.



clientSocket-i do të jetë një socket I familjes AF\_INET me Ipv4 , dhe të dhënat do të dergohen me anë te lidhjes UDP ( connectionless protocol ) .Në addr ruhet dyshja serverName dhe serverPort.



Ne qoftë se fjala e parë është ‘ip’ programi vazhdon tek if-comanda e rradhës , përshndryshe vazhdon me fjaleën ‘nrportit’ e kështu me radhë . Nese gjatëesia është një , ateherë me sendto() dërgojmë kerkesën e enkoduar dhe addr ( i tregon socketit se në qfare IP adrese dhe porti duhet të shkojne të dhenat ) . Nëse gjatësia nuk është njëe printohet mesazhi dhe unaza përsëritet duke kërkuar përsëri një kërkesë . 

Nëse fjala e parë nuk përputhet me asnjë nga shërbimet e serverit printojmë mesazhin “Kërkesa juaj nuk ështe valide” dhe përseri kërkojmë një kërkese të re . Pasi dërgohet kërkesa klienti pret përgjigjjen e serverit përmes metodës recvfrom() dhe në fund e printon atë .

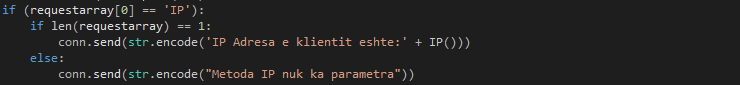


Pas kushtëzimit try : ekzekutohet finally : e cila mbyll socketin e përkohshëm të klientit .

Metodat

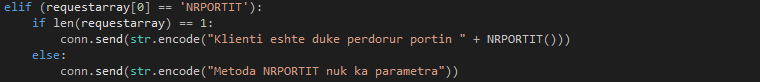
Metodat e përdorura : IP , NRPORTIT , NUMERO , ANASJELLTAS , KOHA , PALINDROM , LOJA , GCF , KONVERTO , HASHPËD , PLATFORM si dhe EXIT .

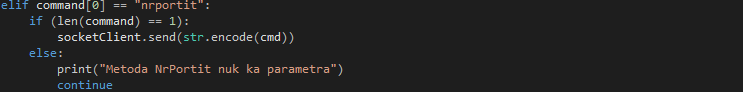
Metoda IP:

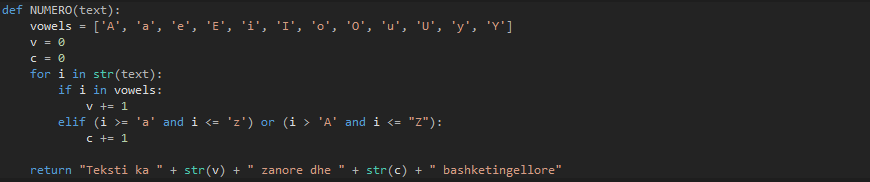
Validimi ne Server:

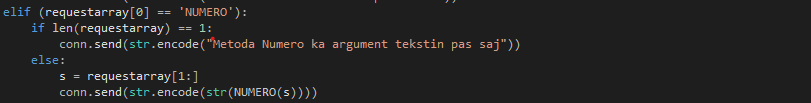
Validimi ne Klient :

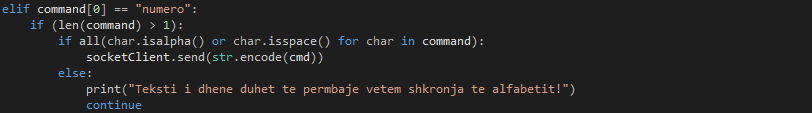
Metoda NRPORTIT:

Validimi ne Server:

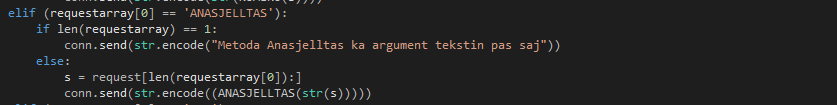
Validimi ne Klient:

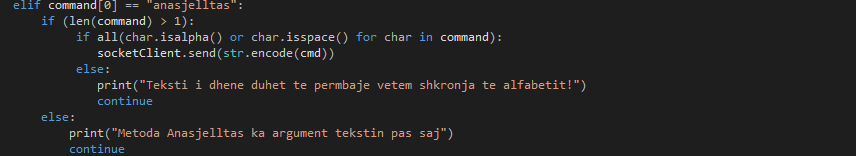
Metoda NUMERO: (Metoda numëro numëron bashtingëlloret dhe zanoret në kërkesën e klientit)

Validimi ne Server:

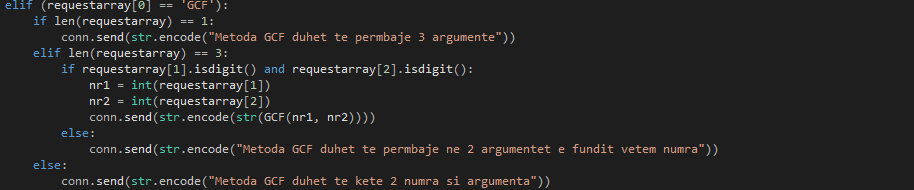
Validimi ne Klient:

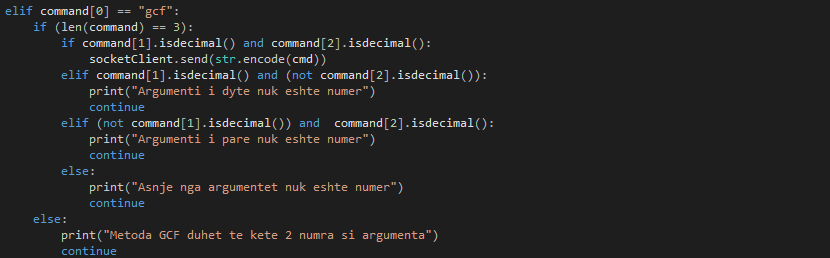
Metoda ANASJELLTAS: (Metoda anasjelltas kthen mbrapsht tekstin e shfrytëzuesit )

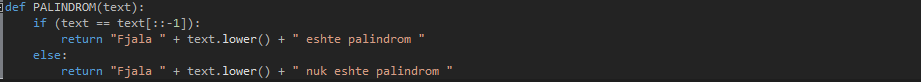
Validimi ne Server:

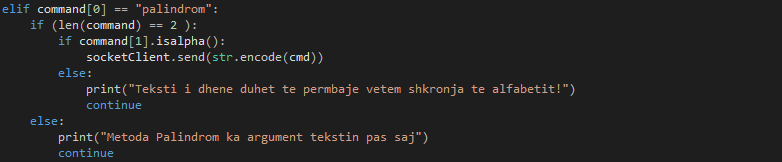
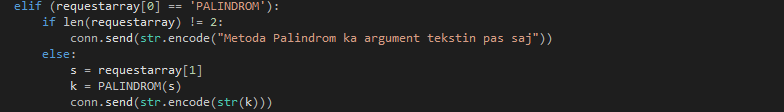
Validimi ne Klient:

Metoda GCF: (Metoda GCF gjen greatest common factor mes dy numrave)

Validimi ne Server:

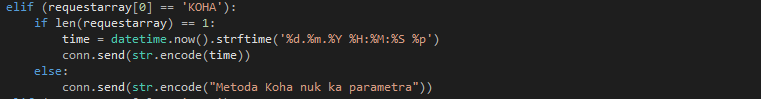
Validimi ne Klient:

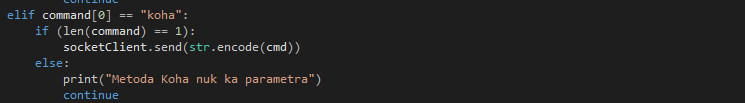
Metoda PALINDROM: (Metoda Palindrom tregon se a është fjala e njejtë edhe nese lexohet nga ana e kundërt)

Validimi ne Server:

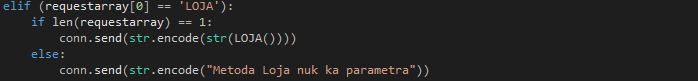
Validimi ne Klient:

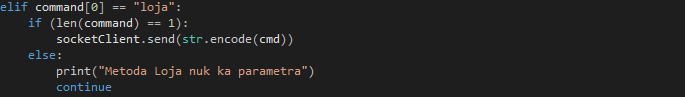
Metoda KOHA: (Metoda Koha kthen kohën ekzakte tek klienti)

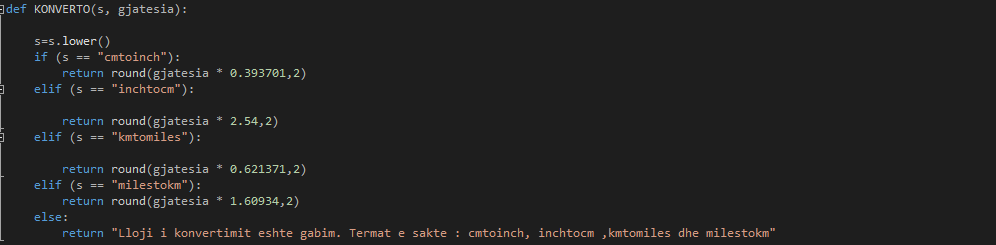
Validimi ne Server: 

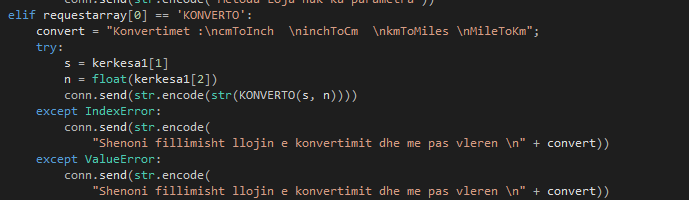
Validimi ne Klient:

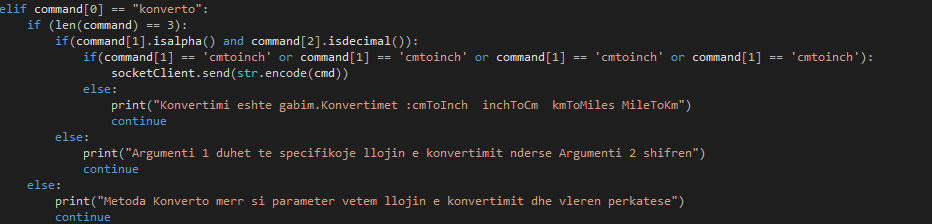
Metoda LOJA: (Metoda loja kthen 5 numra random në range 1-35

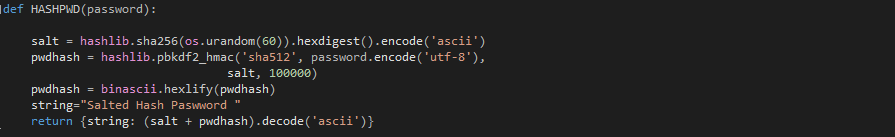
Validimi ne Server: 

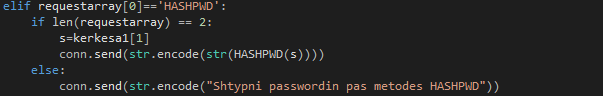
Validimi ne Klient: 

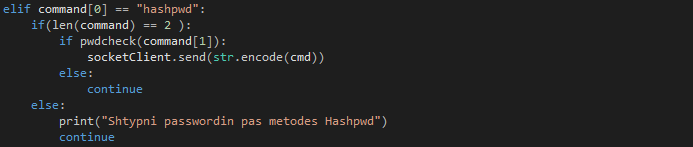
Metoda KONVERTO: (Metoda konverto konverton shifra cmtoinch-inchtocm dhe kmtomiles-milestokm)

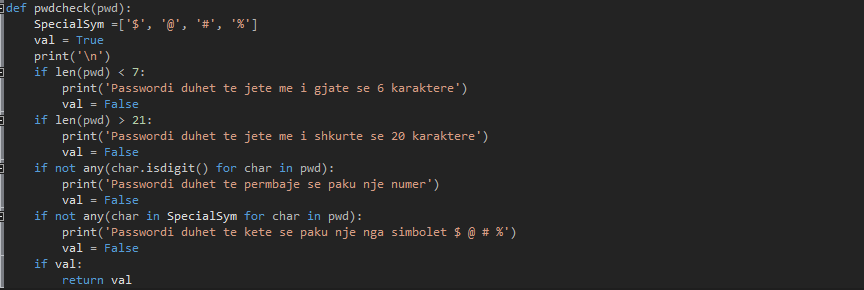
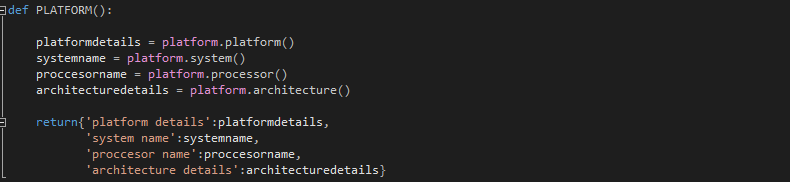
Validimi ne Server: 

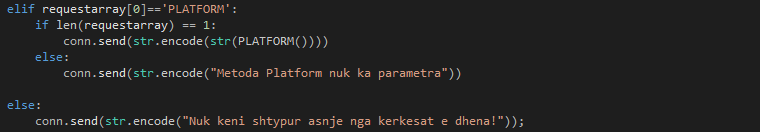
Validimi ne Klient: 

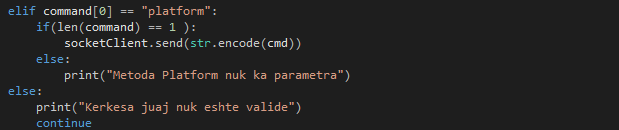
Metoda HASHPWD: (metoda hashpwd kthen salt hash password të klientit) 

Validimi ne Server: 

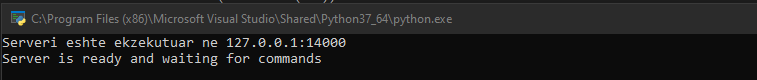
Validimi ne Klient: 

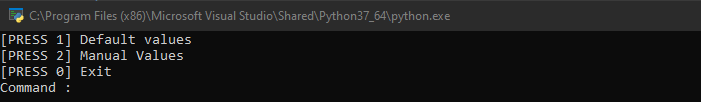
Metoda PLATFORM: (Metoda platform kthen disa detaje pajisjes sëklientit)

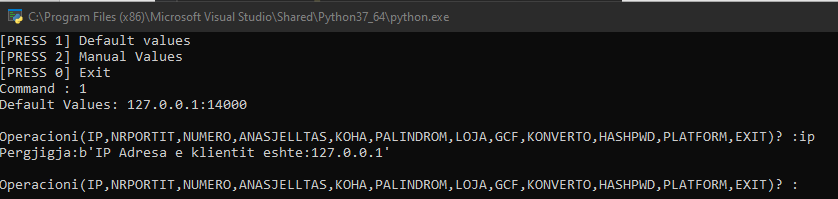
Validimi ne Server: 

Validimi ne Klient:

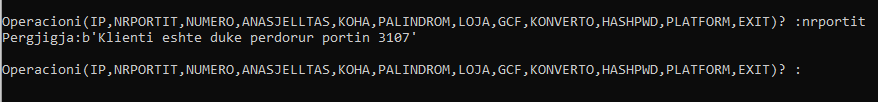
Testimet e metodave

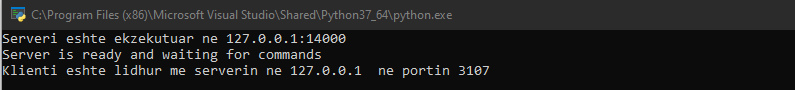
Sapo ekzekutohet serveri do të shfaqet mesazhi në vijim :

Sapo ekzekutohet klienti do të shfaqet mesazhi në vijim :

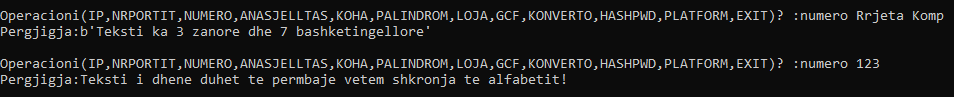
 Do të vazhdojmë me Default values dhe do të testojme metodat si dhe do të japim disa kërkesa gabim per te shikuar se si janë validuar këto metoda .

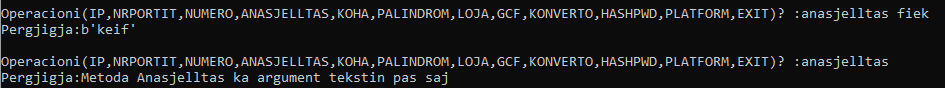
Adresa e klientit është 127.0.0.1

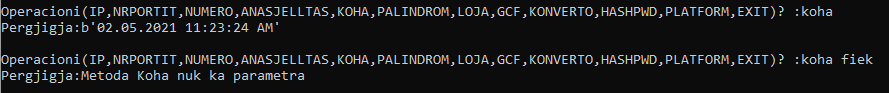


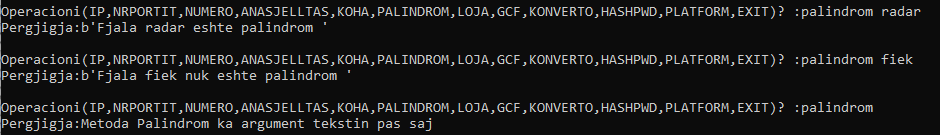
 Klienti është duke perdorur portin 3107 . Ndërkohe mund te shohim se sa here që klienti jep nje kërkesë tek serveri shënohet IP adresa dhe porti . Mund të vërejmë se porti perputhet .

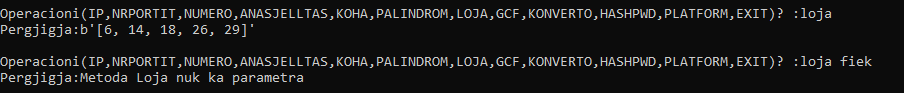
Numëro Rrjeta Komp kthen rezultatin e duhur mirëpo numëro 123 paraqet error .

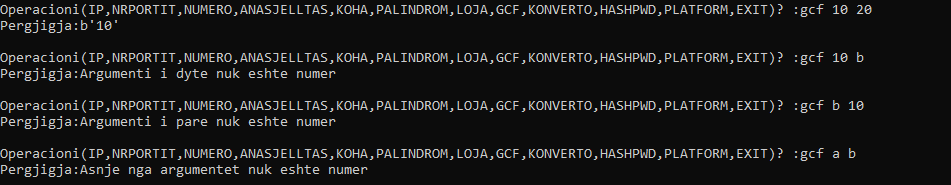


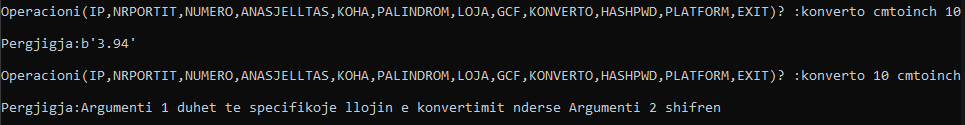
Në qoftë se nuk japim parameter në metodat të cilat nevojiten serveri dërgon këtë mesazh :

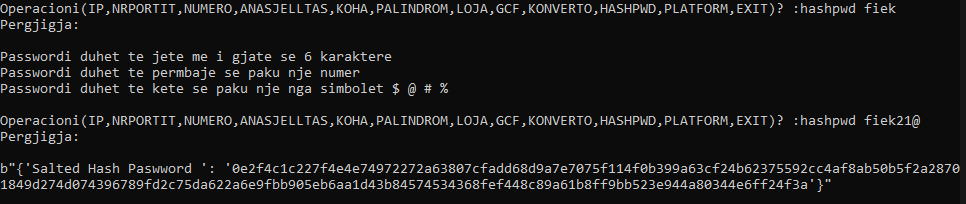
Njëlloj , në metodat ku nuk nevojiten parametra serveri dërgon këtë mesazh :

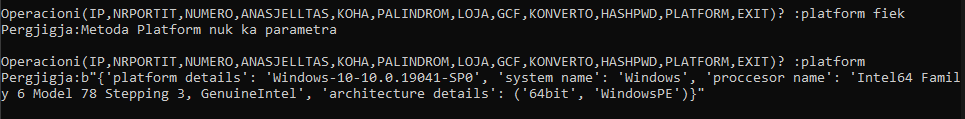
Fjala Radar është palindrom , ndërsa Fiek jo !

Metoda Loja ka kthyer 5 numra random në range 1-35

Metoda GCF ( Greatest comon factor ) ka disa validime me shumë, sic shihet edhe ne testim . Kjo metodë lejon vetëm numra , si dhe I tregon klientit se cili argument nuk është numer .

Metoda konverto merr si parameter llojin e konvertimit dhe më pas shifrën dhe nuk lejon renditje tjetër :

Metoda HASHPWD nuk lejon qe passwordi të jetë i dobët dhe ka kushtëzimet ne vijim . Përndryshe kthen hash passwordin së bashku me salt .

Metoda platform nuk lejon parametra të tjerë . Dhe kthen disa informata per platformen e klientit.

REFERENCAT

<https://www.w3schools.com/python/>

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

<https://stackoverflow.com/>