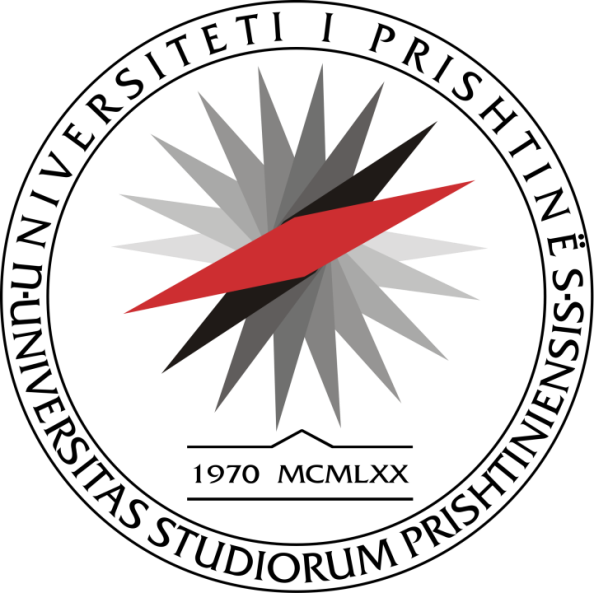
Universiteti i Prishtinës “Hasan Prishtina”  
Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Departamenti i Kompjuterikës



RAPORT: **Dizajnimi KLIENT-SERVER**

LËNDA: Rrjeta Kompjuterike

Profesori i lëndës: Prof. Blerim Rexha Studentja: Teuta Ukshini

Asistenti i lëndës: MSc. Haxhi Lajqi Nr.ID-së: 170714100084

Prishtinë, 2020

Përmbajtja

**Abstrakt**

**Hyrje**

**METODAT**

Përshkrimi i metodave

* Metoda IPADDRESS
* Metoda PORT
* Metoda COUNT
* Metoda REVERSE
* Metoda TIME
* Metoda GAME
* Metoda PALINDROME
* Metoda GCF
* Metoda CONVERT
* Metoda UP\_LOW
* Metoda OPENLINK

**Programimi me Sockets sipas TCP**

**Programimi me Sockets sipas UDP**

**Testimi**

Testimi TCP klient dhe TCP server

Testimi UDP klient dhe UDP server

**Trajtimi i Gabimeve**

**Përmbledhje rreth rezultateve**

**MultiKlienti**

**Referencat**

Abstrakt

Nevoja për ekzistimin e një lidhjeje klient/server në rrjeta kompjuterike ka lindur për arsye se zakonisht klienti ka kapacitet të kufizuar dhe për të bërë punën më të shpejtë dhe efikase kanë hyrë në përdorim serverët të cilët kryejnë çfarëdo veprimi që klienti ka kërkuar dhe kthejnë vetëm rezultatin final tek klienti, pra i gjithë procesi kryhet tek serveri dhe klienti nuk ngarkohet. Kjo i lejon klientit dhe serverit t’i testoj lidhjet e tyre duke komunikuar nëpërmjet socket-ave.

Ekzistojnë dy mënyra të komunikimt të klientit me serverin , lidhja TCP dhe lidhja UDP. Lidhja TCP është lidhje më e sigurtë, serveri krijon lidhje me klienit në fillim, informatat nuk humbin gjatë rrugës dhe klienti informohet për çdo parregullsi , ndërkaq tek lidhja UDP serveri nuk krijon lidhje të vazhdueshme me klientit , shkëmbimi i informatave është i shpejtë por rritet mundësia e humbjes se informatave për shkak mungesës së lidhjes. Në realizimin e këtij projekti janë programuar të dy lidhjet , lidhja TCP dhe UDP. Dallimi mes programimit të këtyre dy programeve nuk është i madh pasi që soketat na mundësojnë të krijojmë soketat e nevojshëm për secilën lidhje. Një server duhet të jetë në gjendje te komunikojë me më shumë se një klient njëkohësisht dhe gjithashtu duhet të merret parasysh se ai duhet të pranojë dhe të kthejë informata tek klienti vazhdimisht dhe për implementimin e këtyre vetive tek serveri kemi perdorur threading modul i cili i mundëson qasjen e shumë klientave njëkohësisht tek i njejti server pa paraqitur problem dhe ngatërresë mes kërkesave.Gjithashtu, përveq kësaj që të mos kemi ndërprerje të punës së serverit edhe pasi që klienti të jep një komandë të panjohur kemi programuar serverin në atë mënyrë që serveri të injoroj kërkesat e panjohura por të mos shkëputet lidhja me klientin.

Hyrje

Ky raport pershkruan ecurinë zhvillimit të detyrës së dhënë në kuadër të lëndës Rrjeta Kompjuterike , përkatësisht projekti i parë. Zakonisht ashtu si në jetën reale gjatë komunikimit nëpër rrjeta ekziston dërguesi(klienti) dhe marrësi(serveri), pra, klienti është ai që vazhdimisht dërgon dhe kërkon të dhëna nga serveri kurse serveri iu përgjigjet atij. Për realizimin e klientit dhe serverit është përdorur gjuha programuese PYTHON dhe editori Visual Studio 2019 ndërsa për krijimin e lidhjes mes klientit dhe serverit janë përdorur soketat(sockets) si pika fundore në secilën anë të cilat mundësojnë qarkullimin e të dhënave mes klientit dhe serverit.

Ky projekt përshkruan dizajnimin, implementimin dhe testimin e komunikimit klient-server përmes socket programming në gjuhën programuese Python. Protokolli i përdorur do të jetë FIEK protokolli i cili u mundëson serverëve dhe klientëve komunikimin përmes dy versioneve të tij, si TCP-protocol dhe si UDPprotocol. Serveri përmban 11 kërkesa (metoda) të cilave mund t’iu referohet klienti. Për secilën kërkesë të klientit server kthen përgjigjen përkatëse mbrapsht te klienti.

Mënyra se si funksionon ky është se programi i krijuar si server përmban disa metoda të cilat janë:IPADRESS,PORT,COUNT,TIME,GAME,REVERSE,PALINDROME,GCF,CONVERT,si dhe metodat shtesë :UP\_ LOW,OPENLINK

Projekti është i zhvilluar në ne 4 programe: TCP Klienti ,TCP Serveri (për lidhjen TCP) dhe UDP Klient , UDP Server(për lidhjen UDP).

Metodat

Këtu bëhet përshkrimi i metodave të caktuara,perkatësisht 11 metodave përkatëse.

**Përshkrimi i metodave**

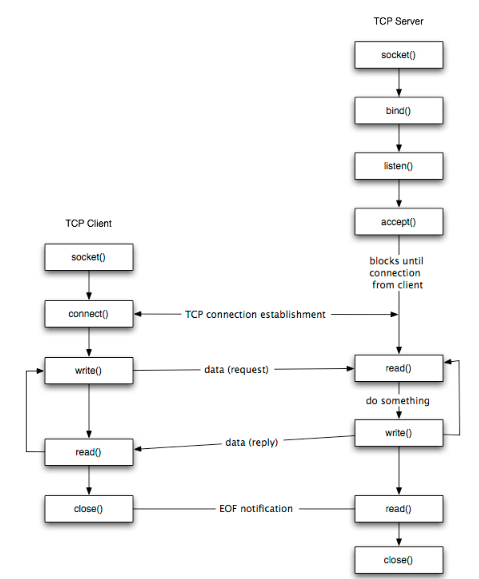
* **Metoda IPADDRESS-** me anë të metodës IPADDRESS klienti është në gjendje të gjej IP e pajisjes së tij. Për të thirrur këtë metodë nuk ka nevojë të shënohet ndonjë argument,por vetëm emri IPADDRESS.
* **Metoda PORT-** kthen portin të cilin klienti është duke e përdorur . Për realizimin e kësaj metode janë përdorur komandat për të dërguar dhe marrë informata prej serverit i cili si përfundim ka kthyer portin e klientit.
* **Metoda COUNT-** Gjen numrin e zanoreve dhe bashtinglloreve në tekstin

e dhënë dhe kthen si përgjigje numrin e tyre. Kjo metodë merr si parametër një tekst të çfarëdoshëm, pastaj krahason secilin karakter të atij stringu me fushën (array) tonë me emrin COUNT, ku janë ruajtur të gjitha zanoret dhe bashtinglloret e alfabetit shqip.. Me përshkimin e gjithë tekstit të caktuar kthehet nga metoda numri i zanoreve dhe të bashtinglloreve në atë tekst.

* **Metoda REVERSE-** Merr si parametër një tekst të caktuar dhe ne fund kthehet teksti i caktuar.
* **Metoda TIME-** kthen kohën aktuale ne server ,e cila është përfituar nga libraritë e importuara ne python.
* **Metoda GAME-**kthen një varg prej 5 numrave random të rangut 1 deri 35.Për realizimin e këssaj metode së pari është importuar libraria e gatshme random.
* **Metoda PALINDROME-** ka për qëllim të gjej nëse teksti i caktuar është apo nuk është palindrome,kjo bëhet me ndërlidhjen tij me metoden reverse pasi që një fjalë të jetë palindrome duhet të ketë të njejtin kuptim edhe mbrapshtë.
* **Metoda GCF-** me anë të kësaj metode gjejme faktorin me te larte të dy parametrave të dhënë.
* **Metoda CONVERT-** Kjo metodë i ofron klientit mundësinë të konvertojë madhësi të caktuara në njësi të ndryshme.Kthen si rezultat konvertimin e parametrit të dhënë varësisht prej opcionit të zgjedhur.Opcionet e konvertimit janë:CmToFeet,FeetToCm,KmToMiles dhe MilesToKm.
* **Metoda UP\_LOW-** Gjen numrin e shkronjave të mëdha dhe të vogla në tekstin e dhënë dhe kthen si përgjigje numrin e tyre. Kjo metodë merr si parametër një tekst të çfarëdoshëm, pastaj krahason secilin karakter të atij stringu me fushën (array) tonë me emrin UP\_LOW, ku janë ruajtur të gjitha shkronjave të mëdha dhe të vogla. Me përshkimin e gjithë tekstit të caktuar kthehet nga metoda numrin e shkronjave të mëdha dhe të vogla në atë tekst.
* **Metoda OPENLINK-**me anë të ksaj metode është zhvilluar thirrja që të bëhët hapja e cfardo linkut ne ëebbroëser me anë të librarisë adekuate.

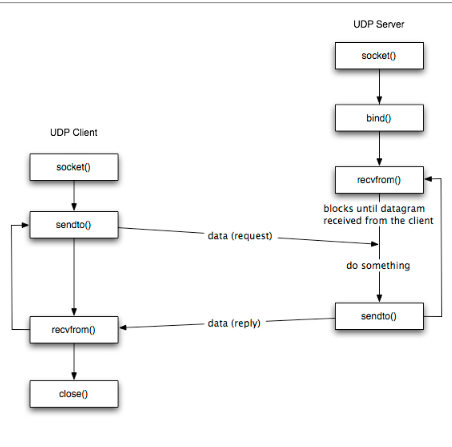
**Programimi me Sockets sipas TCP**

Lidhja TCP ka dy programe TCP Klient dhe TCP Server, ku bëhet komunikimi klient-server, ku dërgon te dhënat pa asnjë humbje, sepse është I besueshëm për shkëmbimin e të dhënave.Pasi që të vendoset lidhja nëse njëra anë dëshiron të dërgoj të dhëna,ajo vetëm vendos të dhënat në lidhjen TCP përmes socketit dhe dërgohen ne anën tjetër.Me vendosjen e kësaj lidhjeje është e mundur të shkëmbehen të dhënat përmes socketave të cilët punojnë sipas parimit të TCP.

TCP Klient-Server

**Programimi me Sockets sipas UDP**

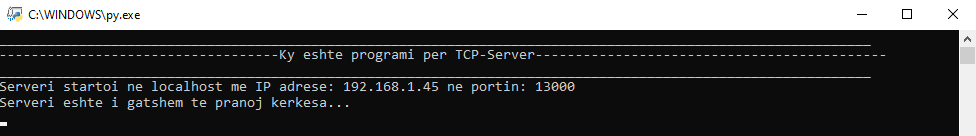
Lidhja UDP serveri nuk krijon lidhje të vazhdueshme me klientit , shkëmbimi i informatave është i shpejtë por rritet mundësia e humbjes se informatave për shkak mungesës së lidhjes. Pra, nuk është i besueshem.



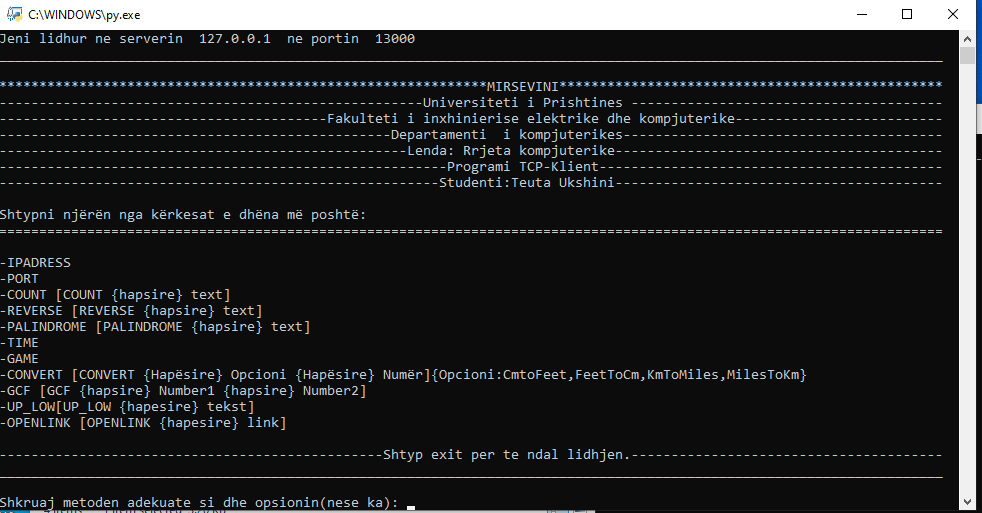
UDP klient server

**Testimi**

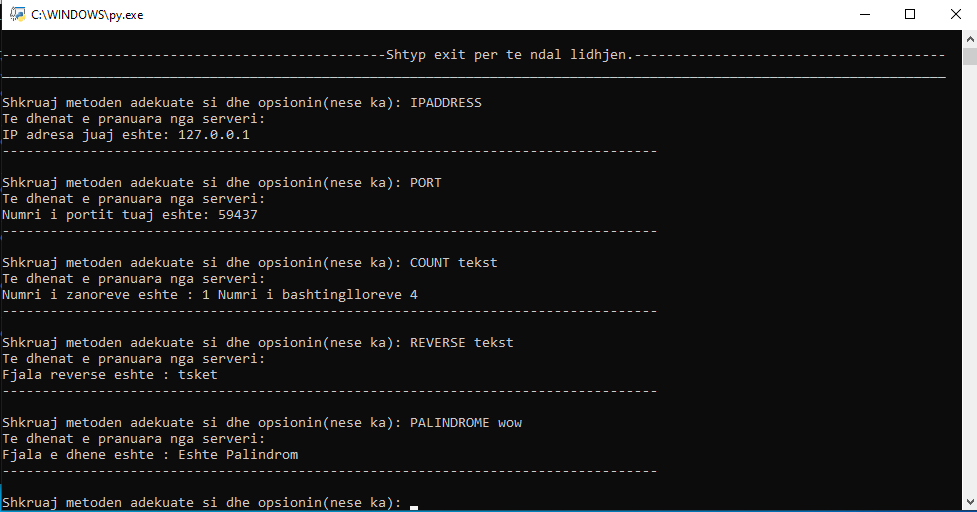
Me hapjen e programit TCP server do te shfaqet kjo dritare:

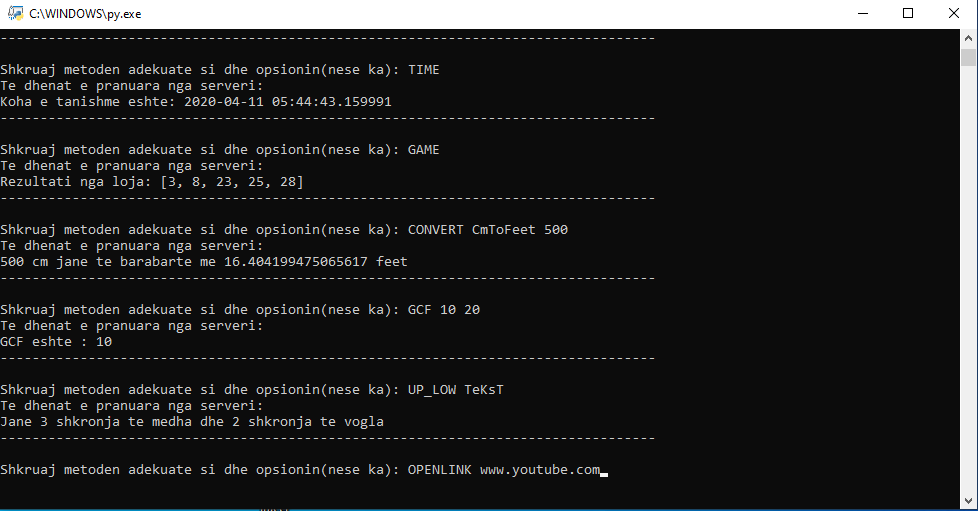


Menjëherë pas hapjes së programit te klientit do të shfaqet një menu , ku tregohen metodat të cilat janë të qasshme dhe përdorimi i tyre:



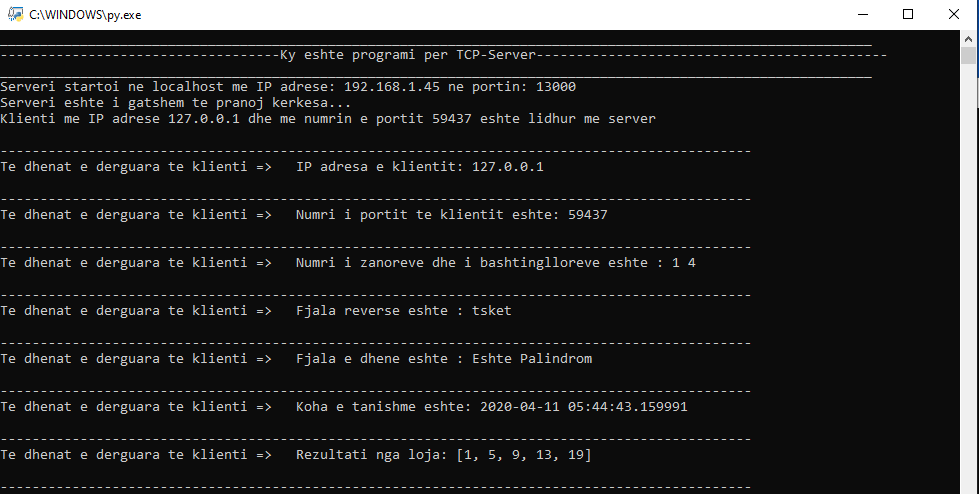
Këtu keni screenshots nga testimi i metodave te TCP Klientit

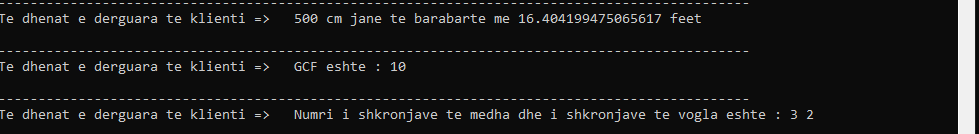




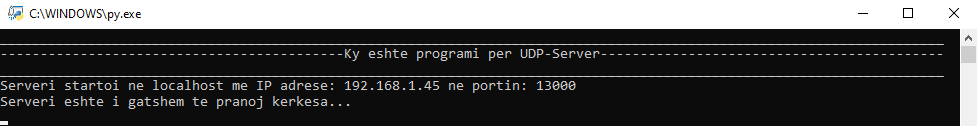
Këtu keni screenshots nga pamjet e testimit te metodave ne programin TCP server

Gjatë komunikimit klient-server apo gjatë dërgimit të përgjigjeve nga serveri tek klienti, në anën e serverit vazhdimisht do të shtypen të gjitha të dhënat që dërgohen te klienti. Këto janë pamjet që shfaqen te programi i serverit.

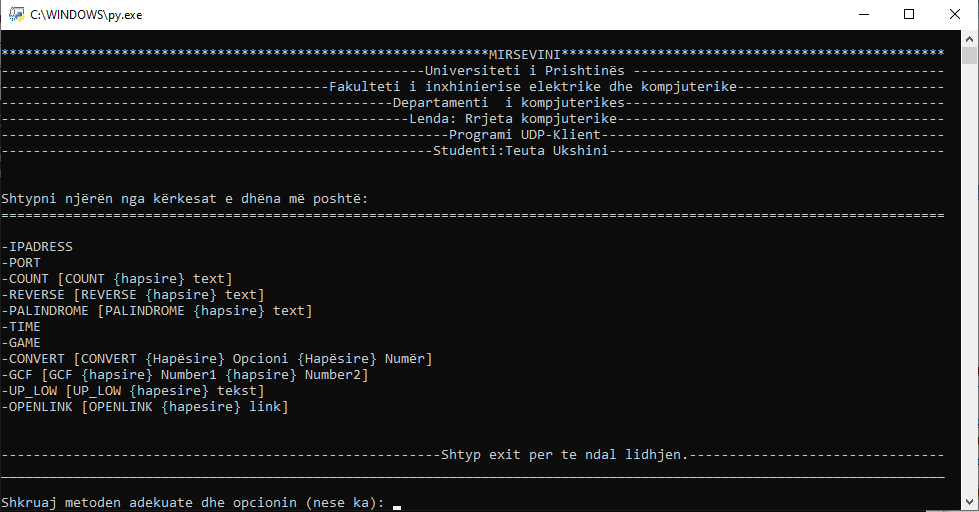




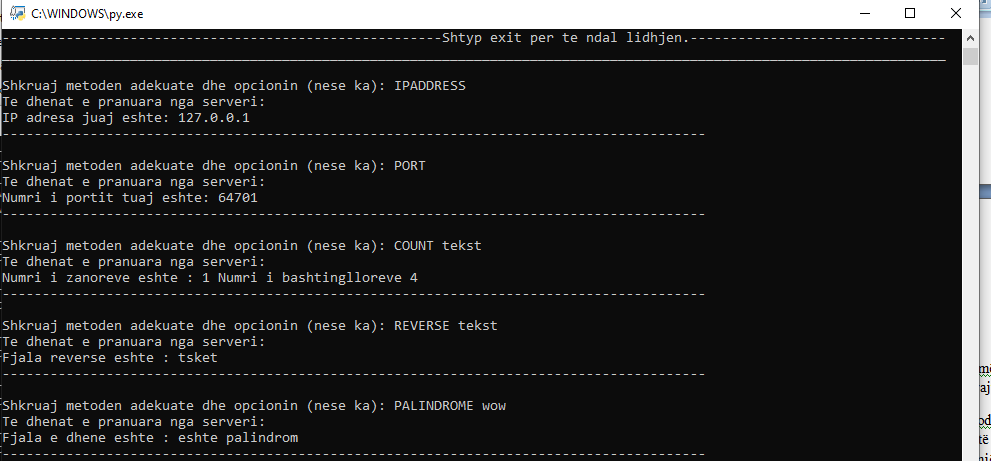
Po ashtu, të njejtat vlejnë po ashtu edhe për UPD

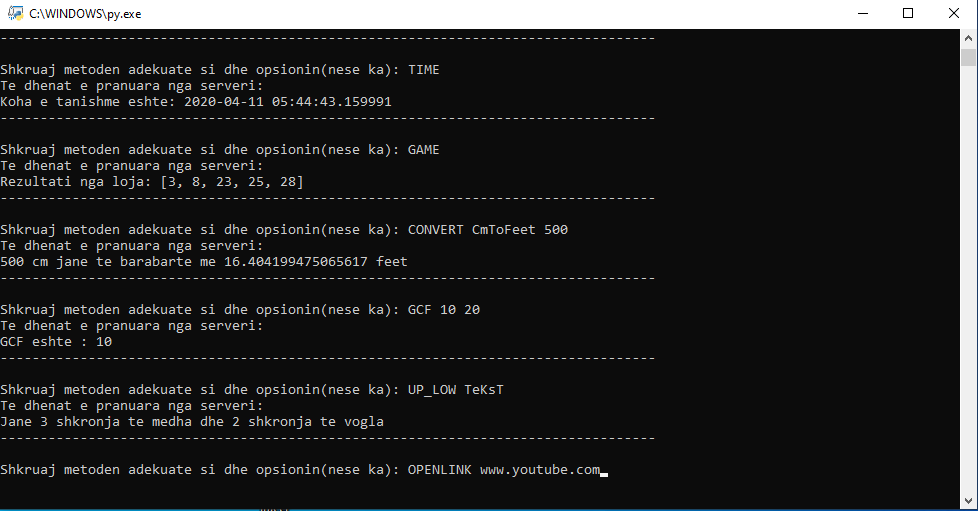


Menjëherë pas hapjes së programit të klientit edhe në UDP klientin klientin do të shfaqet një meny, ku tregohen metodat të cilat janë të qasshme dhe përdorimi i tyre:

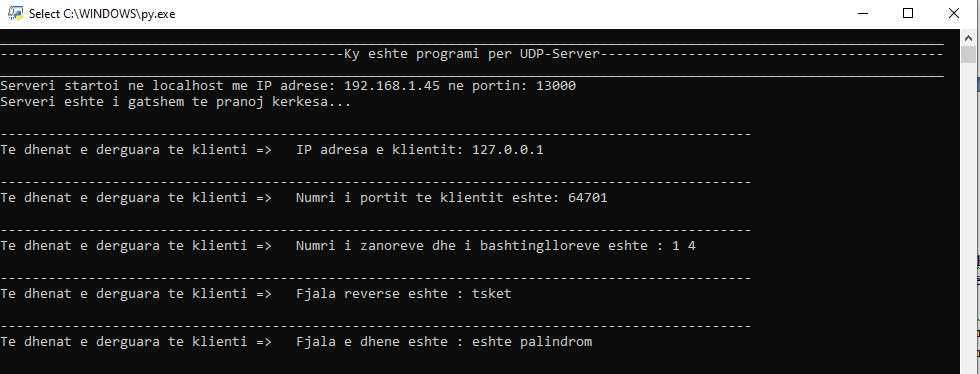


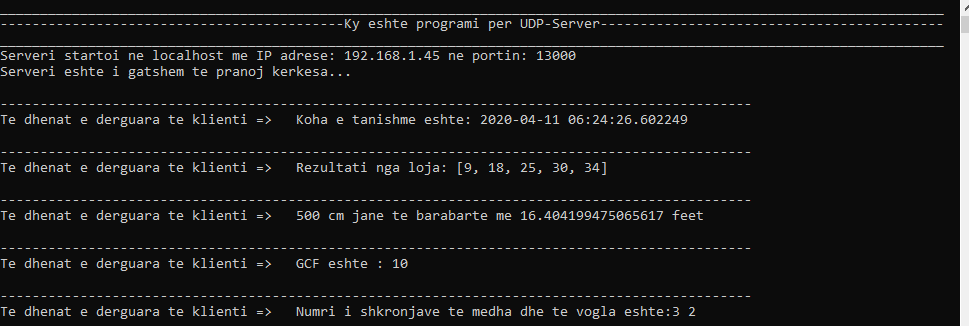
Kurse këto janë screenshots gjatë testimit në programet e UDP klientit





Gjatë komunikimit klient-server apo gjatë dërgimit të përgjigjeve nga serveri tek klienti, në anën e serverit vazhdimisht do të shtypen të gjitha të dhënat që dërgohen te klienti. Këto janë pamjet që shfaqen te programi i serverit 🡪UDP server

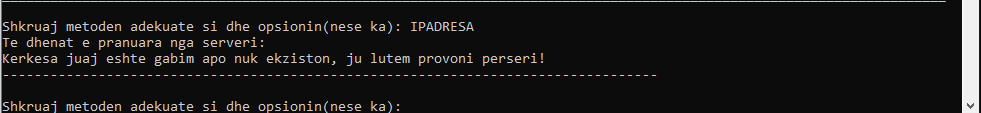




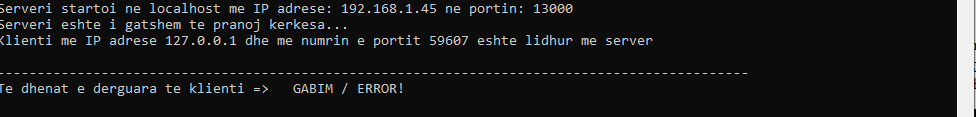
**Trajtimi i Gabimeve**

Gjatë shënimit të kërkesës që përmban emrin e metodës që dëshirojmë mund të bëhen gabime të ndryshme. Programet Server janë programuar në atë mënyrë që të trajtojnë këto gabime.

Në klient



Në sever

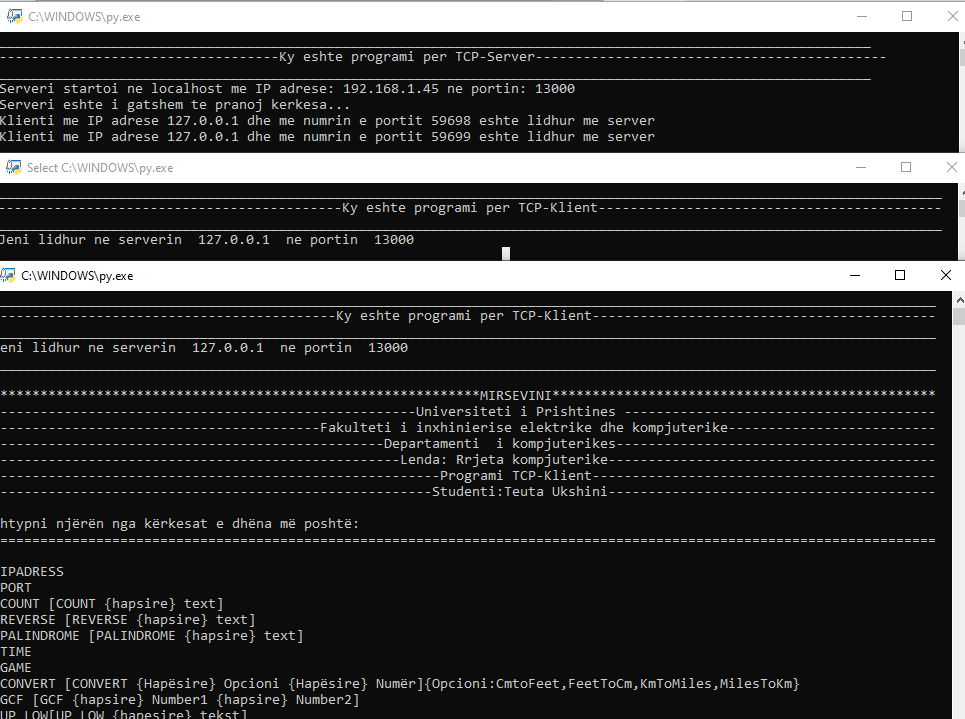


**Permbledhje rreth rezultateve:**

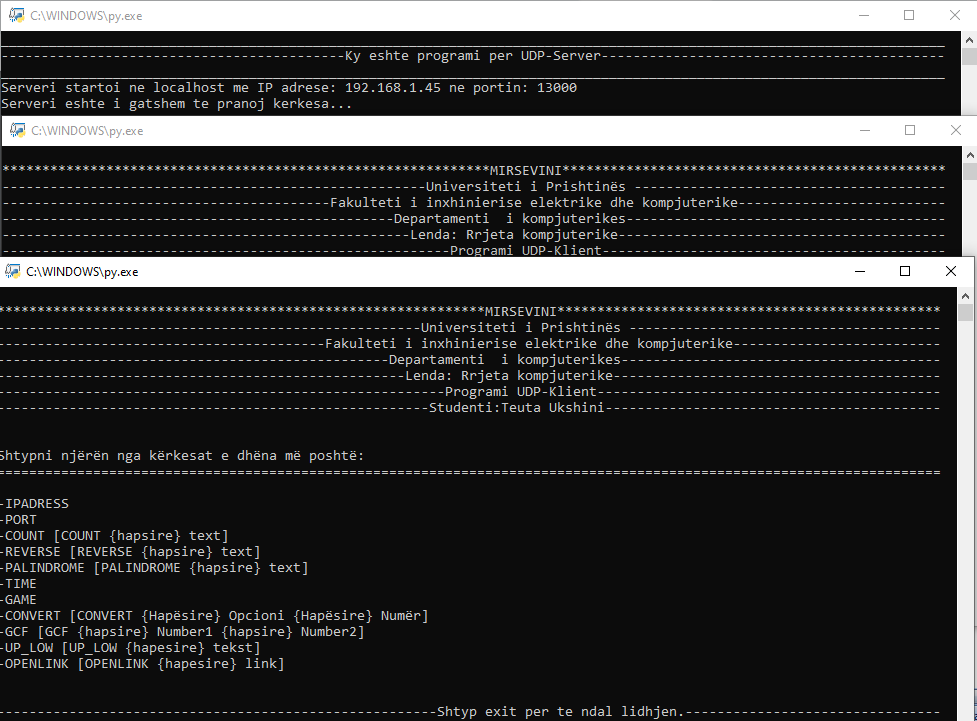
Sic mund të vërehet edhe nga print Screen-at e vendosur me lartë, secila metodë e dhënë funskionon, dhe secila prej tyre kthen rezultatin i cili realisht duhet të kthehet dhe është i saktë.

**MultiKlienti**

**TCP**



**UDP**

****

**Referencat**

* <https://sites.google.com/site/haxhilajqi/rrjetat-kompjuterike-1/ushtrime-rrjeta>
* <https://www.bogotobogo.com/python/python_network_programming_server_client.php>
* <https://www.cs.dartmouth.edu/~campbell/cs60/socketprogramming.html>
* <https://pythontic.com/modules/socket/udp-client-server-example?fbclid=IwAR0OTQlofehGRlXaCtsEOx1ZenWKRfGKdn4O4qkbGTDgQuJsKfuLPKXVPpA>
* <https://www.youtube.com/results?search_query=socket+programming+in+python>