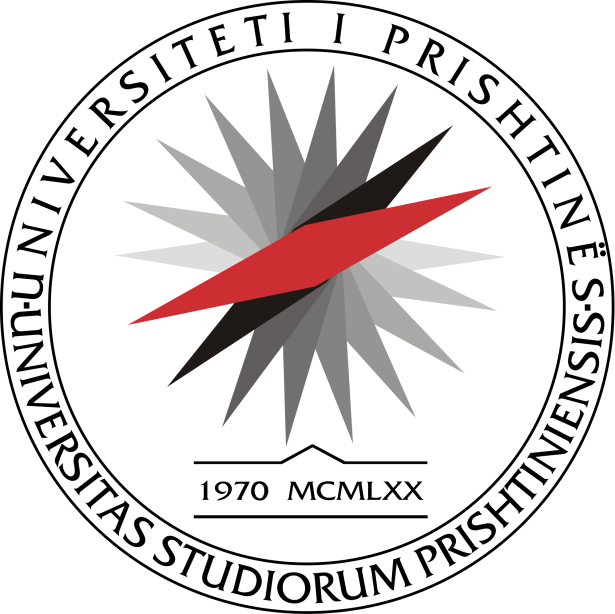
Universiteti i Prishtinës  
 "Hasan Prishtina"

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Departamenti Inxhinieri Kompjuterike



RAPORT MBI PROJEKTIN E PARË NË LENDËN RRJETA KOMPJUTERIKE

TEMA E PROJEKTIT : DIZAJNIMI KLIENT – SERVER

Studentja: Profesori i lendës:  
Blinera Bytyqi Prof.Asoc.Dr.Blerim Rexha

Asistent i lendës:  
 MSc. Haxhi Lajqi

Prishtinë, 2018

*Përmbajtja*

-Hyrje

-Table Of Content

-Pershkrimi I metodave

-Rezultatet e testimit

-Literatura

**Hyrje**

Objektivat e këtij projekti kanë qenë, t’i përforcojmë konceptet dhe protokollet që lidhen me arkitekturen klient server, thirrjet nëpërmjet soketave dhe operimet me soketat TCP dhe UDP.

Qëllimet e tjera të këtij projekti kanë qenë dizajnimi, implementimi dhe testimi i programit klient-server.

Testimi përfshin testimi e TCP klientit dhe UDP klientit me serverin përkatës.

Ky projekt është realizuar në gjuhën programuese Python (version përdorur 3.7.3). server në të dy protokollet ka përdorur hostin “localhost” dhe portin 12000.

**Përshkrimi i metodave**

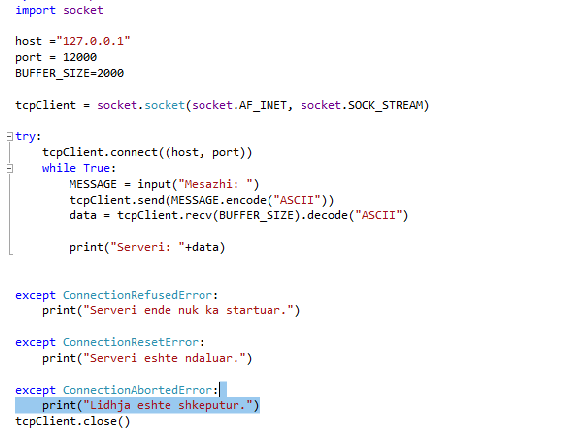
Protokolli FIEK përmban këto metoda: IPADRESA, NUMRIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKOMPJUTERIT, KOHA, LOJA, KONVERTIMI DHE FIBONACCI, ku këto metoda dërgohen nga klienti në server dhe kështu server përgjigjet me një mesazh i cili është specifik për secilën metodë.

*FIEK\_TCP*: Për të bërë një kërkesë klienti FIEK\_TCP duhet të vendosë një lidhje me TCP serverin. Pastaj klienti dërgon kërkesën te server dhe e mbyll lidhjen për dërgim nga klienti në server. Nëse kërkesa është e mundshme, serveri kthen përgjigjen dhe e mbyll lidhjen server klient ose vazhdon me komanë të re.

*FIEK\_UDP*: Për të bërë një kërkesë klienti FIEK\_UDP duhët të dërgojë kërkesën nëpërmjet një UDP datagram për tek server. Nëse kërkesa është e mundshme server kthen përgjigjen në një UDP datagram.

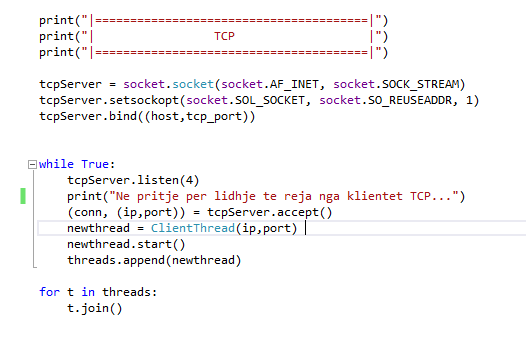
Fillimisht kemi dhene emrin e portit në të cilin duhet të dergojmë të dhena,pastaj kemi krijuar dhe inicializuar një soket me protkollin TCP. Poashtu kemi krijuar lidhje me host dhe e kemi deklaruar një variabël e cila merret nga inputi. Klienti mund të dergojë disa kerkesa që janë disa nga metodat e implementuara në këtë projekt. Pëerderisa kjo lidhje ekziston klient dërgon te dhëna të enkoduara

sipas kodit ASCII kurse serveri I pranon ato te dhena duke I dekoduar sipas te njëjtit kod.



FIEK-TCP-serveri

Në këtë protokoll fillimisht kemi importuar disa librari të gatshme qe na ofron gjuha programuese Python psh import random është librari që na ofron numra të rastësishëm të përdorur në metodën LOJA etj. E kemi krijuar soketin që lidhet me server në portin e dhënë për kërkesat. Serveri pret kerkesën dhe pasi ka lidhje mes serverit dhe klientit ai pranon te dhena.



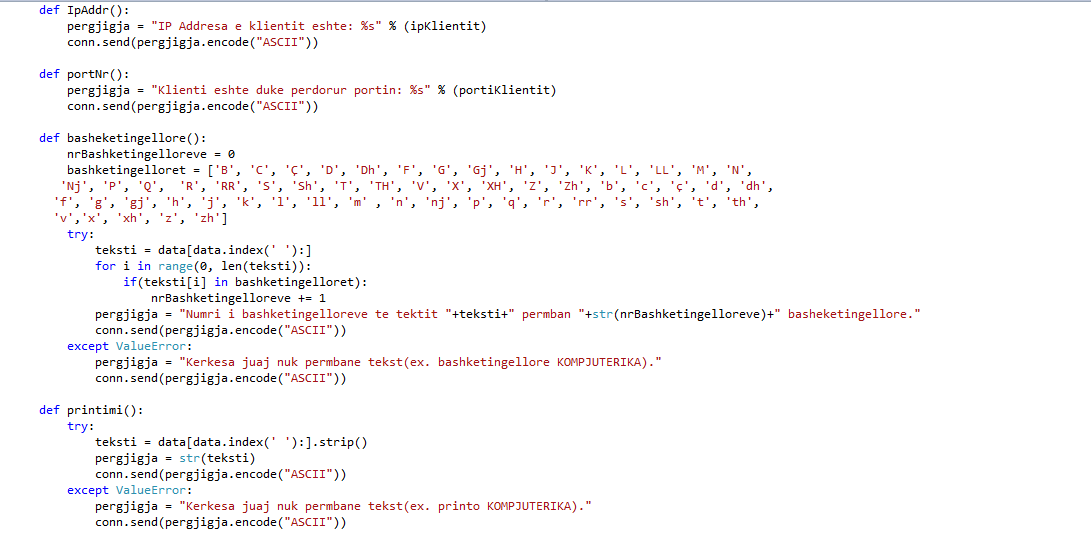
Në pjesën e serverit janë krijuar metodat që mund të thirren në këtë soket dhe në rastin ku server pranon një kërkesë e dekodon dhe pastaj e kthen në linjë kerkesën e enkoduar.

Këerkesat qe mund te hyjnë në ekzekutim jane:

IpAddr – si rezultat paraqet Ip adresën e klientit.

portNr – paraqet numrin e portit të klientit.

Bashketingellore – paraqet numrin e bashkëtingëlloreve ne nje tekst , duke krahasuar qdo shkronjë nga teksti qe pranohet me vargun e zanoreve të dhene në metodë. Në rast se një shkronjë e tekstit është e njejtë me nje shkronjë nga vargu atëherë nr I bashkëtingelloreve që paraqet vlerën mbledhëse te bashkëtingëlloreve (me vlerë fillestare 0) rritet për një.



Printimi – paraqet një fjalë që është dhënë si tekst hyrës nga klienti.

Host – e cila paraqet emrin e hostit

Koha – Paraqet Kohen aktuale në server. Pasi që është importuar nga libraria math atëherë kemi përdorur formatin e gatshëm '%d-%m-%Y %H:%M:%S',d-dita , m-muaji,y-year kurse h-orë,m-minuta,s-sekonda.

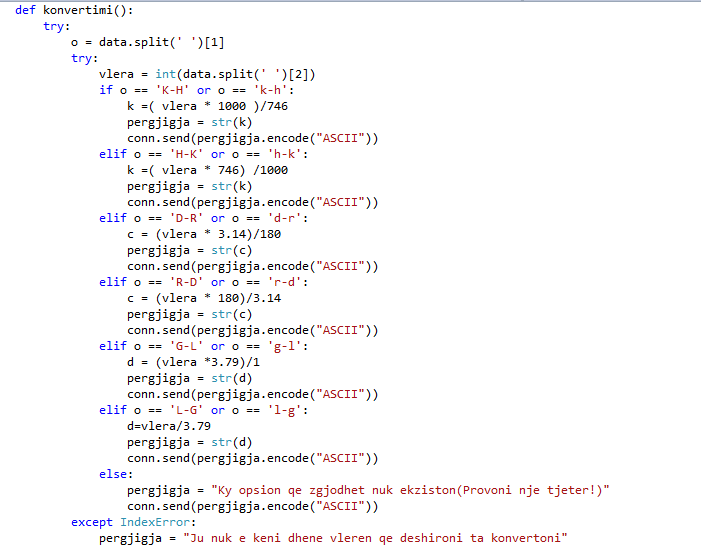
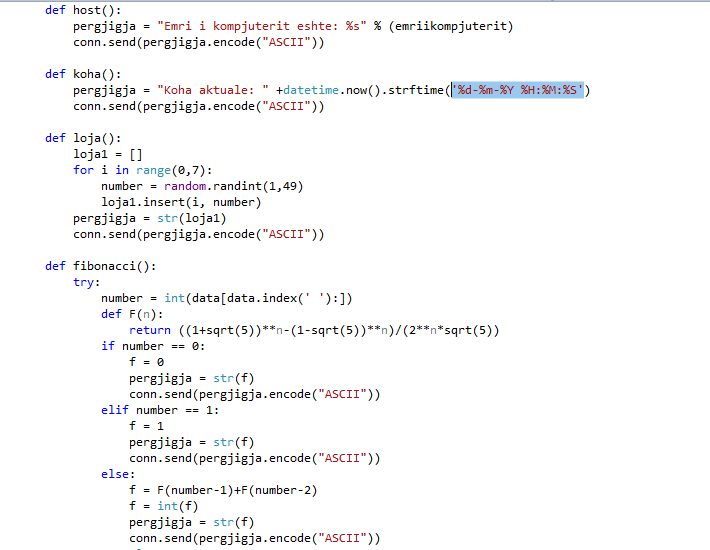
Loja- paraqet 7 numra të rastësishëm nga rangu 1-99 , numrat e rastsishëm Mirren nga libraria random e cila është importuar.

Fibonacci– e cila paraqet numrin FIBONACCI si rezultat i parametrit të dhënë hyrës

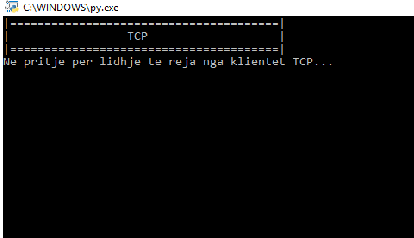
Konvertimi – mundëson konvertimin e një vlere hyrëse te cfarësoshme në njerin nga opsionet:

Kiloëatt-poëer , horsepoëer-Kiloëatt , Degrees – Radian , Radian-Degrees , Galoons-Litres, Litres-Galloons.

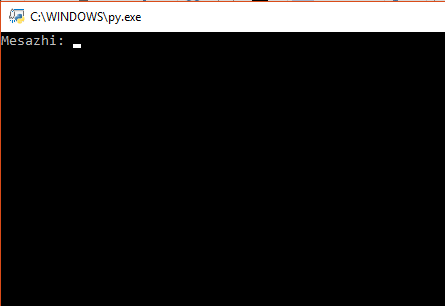
Instructions – na mundësojne sqarime dhe ndihmë në lidhje më metodat , dhe perdorimin e tyre.



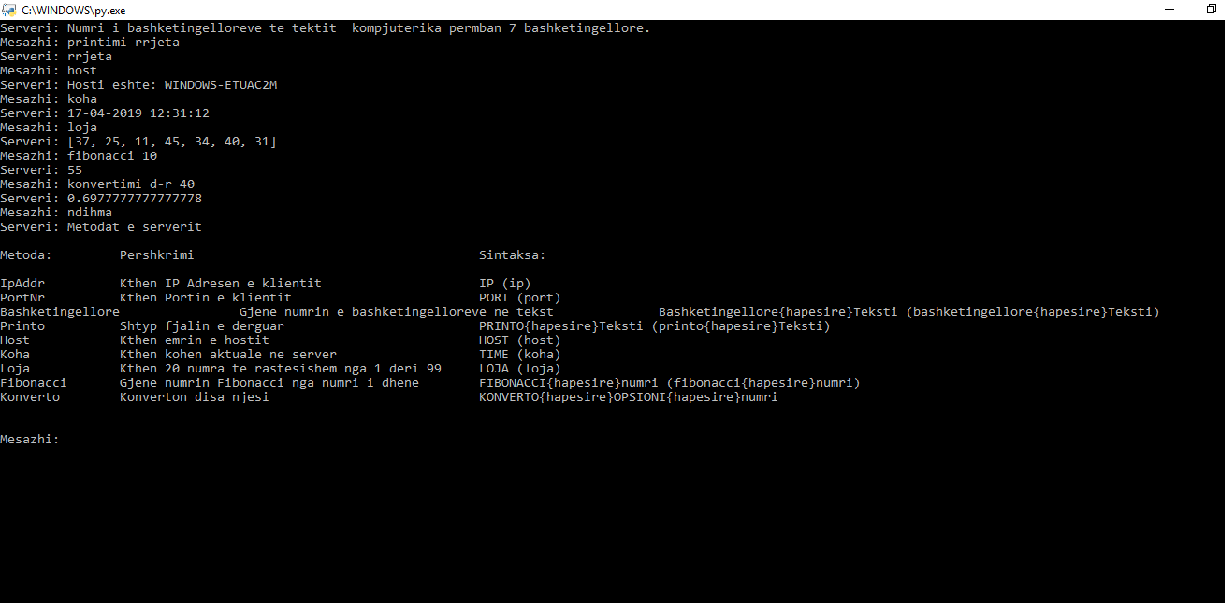
Me ekzekutimin e serverit e fitojme nje rezultat të tille:



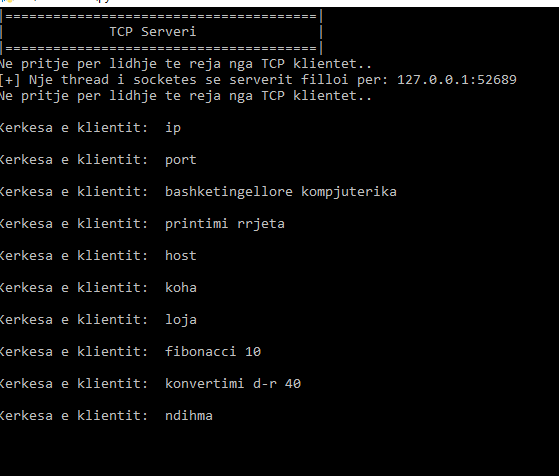
Pasi qe t’a ekzekutojme klientin , kemi :



Varësisht se cfarë na nevojitet , ne mund te shtypim lloje te ndryshme të metodave qe I cekëm më larte:



Nga ana e serverit kemi :



FIEK-UDP-Klienti

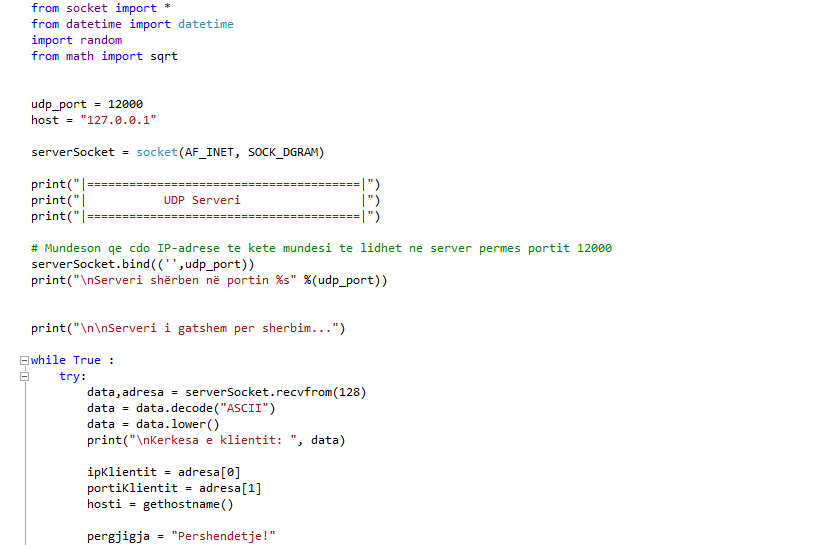
Sikur se tek protokolli TCP edhe ne protokollin UDP krijohet lidhja mes klientit dhe serverit, me dallim se këtu lloji i soketes eshte DATAGRAM , kemi nje lidhje me te shpejte por jo aq te sigurt sikur lidhja TCP. Pra, klienti e dërgon kërkesën vetëm përmes UDP datagram, ku mund të dërgohet vetëm një kërkesë.



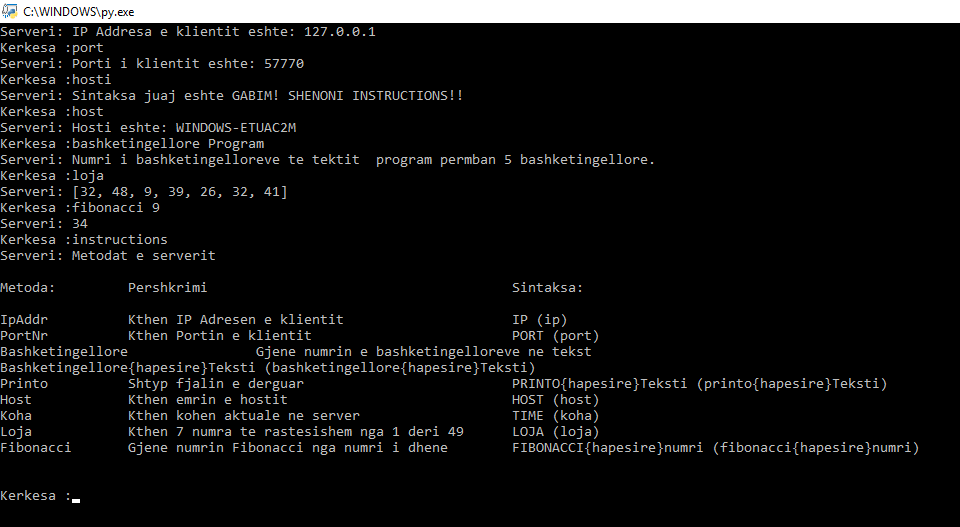
Është përdorur porti i njëjtë për komunikim sikur tek Klienti TCP. Klienti dërgon te dhënën e enkoduar sipas kodit ASCII. Dizajnimi i këtij klienti është i njëjtë me klientin TCP ,pra mund te dërgohen kërkesat e njëjta.

FIEK-UDP-Serveri

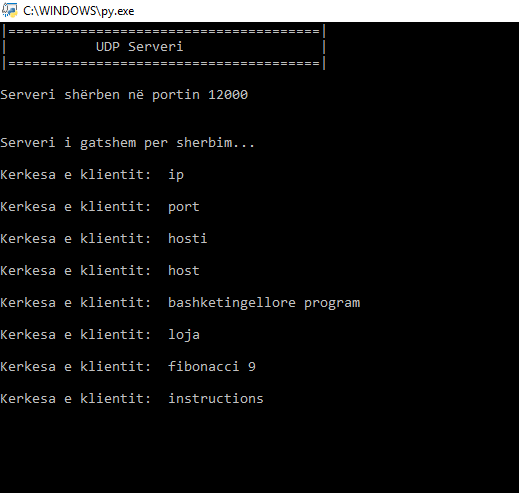
Ashtu sikur klienti UDP qe dallon ne baze te soketit me klientin TCP edhe serverët dallojnë ne baze te tij, pra edhe këtu si tip te soketit zgjedhim SOCK\_DGRAM. Dallimi qëndron edhe ne dërgimin e te dhënave ku përderisa tek Serveri TCP kemi përdorur metodën send tek ky server përdorim metodën sendto dhe ne fund te qdo pergjigjje shtojmë dhe adresën e klientit. Libraritë e importuara janë te njëjta me libraritë e importuara ne Serverin TCP dhe poashtu perdoret edhe porti i njëjtë për komunikim. Nëse kërkesa e dërguar nga klienti është valide , serveri e kthen pergjigjen ne një UDP datagram.



Me ekzekutimin klientit fitojmë:



Me ekzekutimin e serverit :



Literatura

<https://realpython.com/python-sockets/>

<https://www.pubnub.com/blog/socket-programming-in-python-client-server-p2p/>