

UNIVERSITETI I PRISHTINES "HASAN PRISHTINA"
FAKULTETI I INXHINIERISE ELEKTRIKE DHE
KOMPJUTERIKE



RRJETAT KOMPJUTERIKE

PROJEKTI: DIZAJNIMI KLIENT-SERVER

STUDENTI:
OSMAN BYTYQI

PROF.DR:BLERIM REXHA
MSC.HAXHI LAJQI

PRILL, 2021

Përmbledhja

Ky projekt ka të bëjë me protokollet që lidhen me arkitekturën klient/server, thirrjet përmes shtetave dhe operimet me shteta *TCP* dhe *UDP*. Në këtë projekt kemi krijuar protokollin *FIEK* me versionet e tij *FIEK – TCP* dhe *FIEK – UDP*, ku secili nga këta versione ka pjesën e klientit dhe pjesën e serverit. Te protokoli *FIEK* kemi këto funksione: IPADRESSA , NRPORTIT, NUMERO, ANASJELLTAS, PALINDROME, KOHA, LOJA, GCF, KONVERTO, SQRT dhe KOHAEMARRJES të cilat dergohen nga *klineti* tek *serveri*. Serveri përgjigjet specifikisht për secilin funksion dhe injoron qdo kërkesë *jo – valide*

Përmbajta

1	Mjedisi i punes	4
1.0.1	Sistemi Operativ	4
1.0.2	Veglat e punes	4
1.0.3	Funksionet e implemtuara	4
2	Pershkrimi i Funksioneve	5
2.0.1	IP	5
2.0.2	NRPORTI	6
2.0.3	NUMERO	7
2.0.4	ANASJELLTAS	8
2.0.5	PALINDROME	9
2.0.6	KOHA	10
2.0.7	LOJA	11
2.0.8	GCF	12
2.0.9	KONVERTO	13
2.0.10	KOHAEMBERRITJES	14
2.0.11	SQRT	15
3	Konkluzionet	16
4	Burimet e perdorura	16

1 Mjedisi i punes

Qe nje projekt(punim) te filloje dhe ta kryej punen e planifikur eshte e domosdoshme mjedisi i punes. Patjeter qe gjate perpunimit te nje projekti mjedisi i punes ka rendesi i cili na percjell gjate punes sone qe nga fillimi deri ne fund. Ka shume rendesi njohja me hapesiren punuese dhe rolet qe kryejne secili perberes aty. Hapesira punuese perbehet nga keto komponente: sistemet operative, paisjet e ndryshme, qe gjithmone duhet te jene te gatshme per punen qe planifikojme ta bejme; me poshte jane cekur veglat pa te cilat perfundimi i ketij projekti do ishte i pamundur.

1.0.1 Sistemi Operativ

- Ubuntu 20.04.2

1.0.2 Veglat e punes

- Visual Studio Code
- Python 3.7.2
- L^AT_EX

1.0.3 Funksionet e implemtuara

1. IP-Adresa ✓
2. NRPORTIT ✓
3. NUMERO ✓
4. ANASJELLTAS ✓
5. PALINDROME ✓
6. KOHA ✓
7. LOJA ✓
8. GCF✓
9. KONVERTO ✓
10. KOHA E ARRITJES✓
11. SQRT ✓

2 Pershkrimi i Funksioneve

Funksioni eshte nje pjese e veqante e programit (nje nen-program), nje modul qe funksionon i pavarur nga pjesa tjeter e programit duke pranuar parametra hyres dhe kthyer rezultatin e llogaritjes se bere perbrenda tij, ne pjesen e programit e cila ka thirrur funksionin.

Si ne rastin tone qe kemi funksionin *IP* apo *NRPORTIT* te cilet kthejne rezultatet te caktuara si ip-adresen dhe nr e portit te klientit.

2.0.1 IP

```
1 # Funksioni IP
2 def IP(para):
3     return para[0]
4     #return paramteri
```

Consola na dalje:

```
1 #osmani@osmani:~/Desktop/root$
2 Shkruani kerkesen, nese nuk deshironi te vazhdoni shtypni stop: #IP
3 Te dhenat e qe u pranuan nga serveri: # 127.0.0.1
```

1. Kerkesa:

- Klienti kerkon IP-Adresen

2. Aksioni

- Percakton dhe kthen IP adresen e klientit ne forme dhjetore me pike ne nje tekst mesazh

3. Pergjigja:

- Nje mesazh jo me i gjate se 128 karaktere qe permban IP e adreses se klientit
- Mesazhi nuk guxon te permbaje NULL karakter

4. Shtjellimi i Kodit:

- Ne Fillim duhet te startohet serveri dhe me pas klienti, tani vazhdojm duke e shkruar kerkesen ku ne rastin tone eshte: *IP* ose *ip* nese gjate ekzikutimit nuk kemi ndonje problem ather do te paraqitet mesazhi me *IP* e juaj, ne te kunderten do paraqitet ndonje mesazh informues per gabimin per gabimin qe ka nodhur.

2.0.2 NRPORTI

```
1 #Funksioni Nrporti
2 def NRPORTI(para):
3     return para[1]
4     #return parametrin
```

Consola ne dalje:

```
1 #osmani@osmani: ~/Desktop/root$
2 Shkruani kerkesen, nese nuk deshironi te vazhdoni shtypni stop: # NRPORTI
3 Te dhenat e qe u pranuan nga serveri: # 39128
```

1. Kerkesa:

- Numeri i Portit

2. Aksioni:

- Percakton dhe kthen portin e klientit, **Ky duhet te jete porti i klientit dhe jo i serverit!**

3. Pergjigja:

- Nje mesazh jo me i gjate se 128 karaktere qe permban numrin e karaktereve ne tekst.
- Mesazhi nuk guxon te permbaje NULL karakter

4. Shtejllimi i kodit

- Si fillim startohet serveri dhe me pas klienti, tani vazhdojm duke e shkruar kerkesen *NRPOTI* ose *nrporti*. Nese nuk ka ndonje problem gjate dergimit te mesazhit apo kerkeses do te paraqitet mesazhi me portin ne te cilin jeni, ne te kunderten do te paraqitet nje mesazh informues ne lidhje me gabimin.

2.0.3 NUMERO

```

1 #Fun Numero
2 def NUMERO(para):
3     zanoret = 0
4     bashketingelloret = 0
5     for z in para:
6         if z in ['a','e','e','i','o','u','y','A','E','E','I','O','U','Y']:
7             zanoret += 1
8         elif z in ['b','c','c','d','f','g','h','j','k','l','m','n','p','q','r',
9                     's','t','v','x','z',
10                    'B','C','c','D','F','G','H','J','K','L','M','N','P','Q','R',
11                    'S','T','V','X','Z']:
12             bashketingelloret += 1 #inkremnti
13     x = "Zanore :" + str(zanoret) #inkremnti
14     y = "Bashketingellore : " + str(bashketingelloret)
15     return x, y
16     # dhe ne fund return x dhe y pas inkremntit

```

Consola ne dalje:

```

1 #osmani@osmani:~/Desktop/root$
2 Shkruani kerkesen, nese nuk deshironi te vazhdoni shtypni stop:# NUMERO osman
3 Te dhenat e qe u pranuan nga serveri: # ('Zanore :2', 'Bashketingellore : 3')

```

1. Kerkesa:

- Zanoret dhe Bashktingllore

2. Aksioni

- Gjen numerin e zanoreve dhe bashktinglloreve dhe kthen pergjigjen

3. Pergjigja:

- Nje mesazh jo me i gjate se 128 karaktere qe permban numrin e karaktereve ne tekst
- Mesazhi nuk guxon te ket *NULL* karakter

4. Shtjellimi i kodit:

- Pike se pari startojm serverin, keshtu qe vazhdojm te klienti duke e shkruar *NUMERO* funksionin, nese nuk ka ndonje problem gjate dergimit te mesazhit do te paraqitet mesazhi me numerin e zanoreve dhe bashktinglloreve per fjaline te cilen e shkruan klienti, por ne te kunderten do te paraqitet nje mesazh me per gabimin qe ka ndodhur.

2.0.4 ANASJELLTAS

```

1 #funksoni anasjelltas
2 def ANASJELLTAS(para):
3     fjalw = ""
4     for i in para:
5         fjala = i +fjala
6     return fjala
7     #return fjalen e cila bohet reverse

```

Consola ne dalje:

```

1 #osmani@osmani:~/Desktop/root$
2 Shkruani kerkesen, nese nuk deshironi te vazhdoni shtypni stop: # ANASJELLTAS OSMAN
3 Te dhenat e qe u pranuan nga serveri: # NAMSO

```

1. Kerkesa:

- Anasjelltas

2. Aksioni:

- Kthen fjalinë e shtypur ne tekst, hapesirat ne fillim dhe ne fund te fjalisë nuk duhet te kthehen

3. Pergjigja:

- Tekst

4. Shtejllimi i Kodit:

- Gjate thirrjese se funksionit *ANASJELLTAS* dhe fjales qe deshironi te boni reverse athe- here te klineti do te paraqitet ajo fjali vetem ne menyre reverse pra me renditje te kundert do te shfaqet fjala e dhe nga klinetit edhe ketu per qdo gabim qe ndodh gjate ketij procesi serveri do e njoftoj klinetin ne lidhje me gabimin e ndodhur.

2.0.5 PALINDROME

```
1 #funksioni Palindrome
2 if (para == para[::-1]): # :: :paraqet fillimin :mbarimin
3     x = "Teksti eshte polindrom"
4 else:
5     x = "teksti nuk eshte polindrom"
6 y = str(x) #vlera e y e kthejm ne string
7 return y
```

Consola ne dalje:

```
1 #osmani@osmani:~/Desktop/root$
2 Shkruani kerkesen, nese nuk deshironi te vazhdoni shtypni stop: # PALINDROME OKEJ
3 Te dhenat e qe u pranuan nga serveri: # teksti nuk eshte polindrom
```

1. Kerkesa:

- Palindrome tekst

2. Aksioni:

- Kerkon nje fjali dhe tregon a eshte palidrome a jo

3. Pergjigja:

- Serveri tregon fjala e dhene a eshte palindrome

4. Shtjellmi i Kodit:

- Kur e therrasim funksionin *PALINDROM* eshte e domosdoshme qe ta shkruajm edhe nje fjale si parameter dhe pas kesaj serveri na kthen pergjigje por nese gjate kesaj rruge ka nodhur ndonje gabim athere serveri do na kthej nje mesazh informues ne lidhje me gabimin qe ka ndodhur.

2.0.6 KOHA

```

1 #funksioni Koha i cili na kthen kohen aktuale
2 from datetime import datetime #importojm nga librarite e gatshme
3 def KOHA():
4
5     x = datetime.now()
6     y = "Koha aktuale ne pasijen tuaj eshte : " + str(x)
7     return y

```

Consola na dalje:

```

1 #osmani@osmani:~/Desktop/root$
2 Shkruani kerkesen, nese nuk deshironi te vazhdoni shtypni stop:# KOHA
3 Te dhenat nga serveri:#Koha aktuale ne pasijen tuaj eshte :2021-05-01 14:38:05.756780

```

1. Kerkesa:

- Koha

2. Aksioni:

- Percakton kohen aktuale ne server dhe e dergon ate tek klienti si format te lexueshme per klientin

3. Pergjigja:

- Kthen nje pergjigje e cila permban kohen dhe daten

4. Shtjellimi i kodit:

- Si fillim e startojm serverin dhe me pas edhe klientin, mjafton te shkruhet nga ana e klientit *KOHA* apo *koha* dhe ne ekran do te shfaqet mesazhi qe tregon kohen aktuale kjo ndodh duke i importuar librarite e gatshme si ne rastin tone *datetime* qe e kemi importuar *fromdatetime* , por ne qofte se nodh ndonje gabim gjate ketij procesi athere e kemi mesazhin informues mbi detajet e gabimit qe ka ndodhur.

2.0.7 LOJA

```
1 #fun loja
2 def LOJA():
3     x = random.sample(range(1, 35), 5) #random e kemi importuar nga librarite e gatshme
4     y = "Numra te cfaredoshem ne rangun [1,35] : " + str(x)
5     return y
```

Consola ne dalje:

```
1 #osmani@osmani:~/Desktop/root$
2 Shkruani kerkesen, nese nuk deshironi te vazhdoni shtypni stop:# LOJA
3 Te dhenat nga serveri:# Numra te cfaredoshem ne rangun [1,35] : [15, 9, 31, 16, 3]
```

1. Kerkesa:

- Loja

2. Aksioni:

- Kthen 5 numra nga rangun [1,35]

3. Pergjigja:

- Tekst p.sh(1,3,13,17,33)

4. Shjtellmi i kodit:

- Si te qdo funksion tjter edhe tek ky funksion ne fillim e startojm serverin, e shkruajm *LOJA* apo *loja* dhe permes *random* qe e kemi importuar nga librarite e gatshme te *pythonit* ne pergjigje do te shfaqen 5 numera te ndryshem ndermejt rangut 1 dhe 35, por nese diqka shkon jo ne menyren me te mire athere do te njoftohet klienti ne lidhje me qdo gabim qe ka ndodhur.

2.0.8 GCF

```
1 # funksioni qe ngjen faktoren me te madh te perbashket
2 def GCF(a, b):
3     if (b == 0):
4         return a
5     else:
6         return GCF(b, a % b)
7     #% moduli apo mbetja nga pjestimi
```

Consola ne dalje

```
1 #osmani@osmani:~/Desktop/root$
2 Shkruani kerkesen, nese nuk deshironi te vazhdoni shtypni stop:# GCF 3 5
3 Te dhenat e qe u pranuan nga serveri: # 1
```

1. Kerkesa:

- GCF Numër1 Numër2

2. Aksioni:

- Gjen faktoren me te madh te perbashket ne mes dy numrave qe klienti jep.

3. Shtjellimi i kodit:

- Pranojm 2 parametra nga ana e klientit qe ne kete rastin tone ja dy numera dhe atehere funksioni qe e kemi krijuar do e gjej faktoren me te madhe te atyre numerave, por edhe nese gjithqka nuk shkon ne menyren me te mire atehere klienti do jete i informuar prej anes se serverit.

2.0.9 KONVERTO

```

1 #fun konvertuesi
2 def KONVERTO(a, b):
3     if a == "CMTOINCH": #cm->inch
4         x = b/2.54
5         y = "" + str(x) + "in"
6         return y
7     elif a == "INCHTOCM":
8         x = b *2.54
9         y = "" + str(x) + "cm"
10        return y
11    elif a == "KMTOMILES":
12        x = b/1.609
13        y = "" + str(x) + "miles"
14        return y
15    elif a == "MILESTOKM":
16        x = b *1.6091
17        y = "" + str(x) + "km"
18        return y
19    #return y ne fund

```

Consola ne dalje:

```

1 #osmani@osmani:~/Desktop/root$
2 Shkruani kerkesen, nese nuk deshironi te vazhdoni shtypni stop: # konvertto cmttoinch 2
3 Te dhenat e qe u pranuan nga serveri:# 0.7874015748031495inch

```

1. Kerkesa:

- Konvertimi

2. Aksioni:

- Kthen si rezultat konvertimin e opcioneve varesisht opcionit te zgjedhur *cmNeInch*, *inchNeCm*, *kmNeMiles*, *mileNeKm*

3. Pergjigja:

- Nese konvertojm *cmNeInch* 10 kthen rezultatin *3.94inch*

4. Shtjellmi i kodit:

- Si fillim shkruhet startohet serveri dhe mw pas klienti. Mw pas shkruhet kerkesa *konvertimi* ose *KONVERTIMI*, pas hapsires shkruhet ne qka do te kthehet (konvertohet) dhe numri se qfar veprimi do te marresh. Nese nuk ka ndonje problem gjat dergimit te mesazhit, do te paraqitet mesazhi me rezultatin e konvertuar sipas metodes konvertimi. Nese ndodh ndonje gabim do te kthej pergjigjet mbi informacionin e gabimit.

2.0.10 KOHAEMBERRITJES

```

1 # funksioni i cili llogarite shpejtesine per 50km/h
2 def KOHAEMBERRITJES(num):
3     num = float(num)
4     v = 50000/3600
5     t = num / v
6     pergjigja = "Per shpejtesi 50 km/h dhe per rrugen " + str(num) + " metra koha e
7                 mberritjes eshte " + str(t)+ " sekonda"
8
9     return pergjigja
#return pergjigja e cila eshte e barabart me shpjetesine e llog

```

Consola ne dalje:

```

1 #osmani@osmani:~/Desktop/root$
2 Shkruani kerkesen, nese nuk deshironi te vazhdoni shtypni stop: #KOHAEMBERRITJES 19
3 Te dhenatnga serveri:#Per shpejtesi 50 km/h dhe per rrugen 19.0 metra koha e mberritjes
    eshte 1.3679 sekonda

```

1. Kerkesa:

- Koha e mberritjes

2. Aksioni:

- Gjate thirrjes se funksionit si rezultat i parametrut qe eshte by default metra fitojm fitojm shpjetesine per ne sek

3. Shtjellimi i kodit:

- 11. Si fillim shkruhet startohet serveri dhe me pas klienti. Me pas shkruhet kerkesa *KOHAEMBERRITJES*. Nese nuk ka ndonje problem gjat dergimit te mesazhit, do tw paraqitet mesazhi me rezultatin e konvertuar sipas *KOHAEMBERRITJES*. i cili na jep shumen e fituar, ku llogarite qe per shpejtesi $50 \frac{km}{h}$ dhe per rrugen qe e ka japur klienti sa do te jete koha e shprehur ne *sek*.

2.0.11 Sqrt

```

1 #SQRT funksioni
2 def Sqrt(numri) :
3     num = float(numri)
4     num_sqrt = num ** 0.5
5     pergjigja = "Rrenja katrore: " + str(num_sqrt)
6     return pergjigja
7 #return pergjigja e cila na jep rrejen katrore te nje numri

```

Consola ne dalje:

```

1 #osmani@osmani:~/Desktop/root$
2 Shkruani kerkesen, nese nuk deshironi te vazhdoni shtypni stop:# Sqrt 4
3 Te dhenat e qe u pranuan nga serveri: # Rrenja katrore: 2.0

```

1. Kerksa:

- *Sqrt*

2. Aksioni:

- *Sqrt* funksioni qe pret te shkruhet nje numer

3. Pergjigja:

- Numër i plotë kthehet ne rrenje katrore p.sh numeri 9 kthen si rezultat 3

4. Shtjellmi i kodit:

- Si fillim shkruhet startohet serveri dhe me pas klienti.Me pas shkruhet kerkesa *Sqrt* funksioni. Nwse nuk ka ndonje problem gjat dergimit te mesazhit, do te paraqitet mesazhi me rezultatin e konvertuar sipas *Sqrt* funksioni.

3 Konkluzionet

Gjate punimit te ketij projekti kam kuptuar mire protokollet si *TCP/UDP* dhe operimin e tyre permes socketave, menyren se si lidhet nje server me klient, portet, thread-at, disa librari te gatshme te klasave te pythonit gje per te cilat nuk isha i informuar me pare e shume gjera te reja. Sfida e madhe gjate kesaj pune ishte gjuha *python* edhe pse nuk ishte hera ime e pare me kete gjuhe por per shkak te mungeses se `{}` dhe `()` e qe konsideroj si te "mete" te kesaj gjuhe me ka shkatuar pengesa gjate punes, por nese do te ekzistonin ato simbole besoj qe do ishte me e lehte puna.

4 Burimet e perdorura

Krahas asaj qe eshte na ofruar ne ushtrime dhe ne ligjerata kemi perdorur edhe keto burime:

- <http://stackoverflow.com/>
- <https://www.w3schools.com/>
- <https://www.youtube.com/>