

Universiteti i Prishtinës

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Lënda: Rrjetat Kompjuterike – 2020



Profesor i lendes: Blerim Rexha

Asistenti: Msc. Haxhi Lajqi

Studenti: Lirim Beka

ID: 180714100157

Data: 20/04/2020

Permbajtja

1.Veglat e perdorura, versionet e tyre dhe lista e metodave te implementuara

2.Hyrja

3.Pershkrim I kodit te aplikacionit te serverit (TCP dhe UDP)

4.Pershkrimi I kodit te aplikacionit te clientit (TCP dhe UDP)

5.Pershkrimi I metodave

6.Rezultatet e testimeve

1. Veglat e perdorura, versionet e tyre dhe lista e metodave te implementuara

Sistemi operativ:

- Linux

Vegla e perdorur:

- Pycharm

Librarite e perdorura:

- socket
- random
- numpy
- datetime
- _thread
- gcd

Metodat e caktuara nga Assistenti I lendes:

- IPADDRESS
- PORT
- COUNT
- REVERSE
- PALINDROME
- TIME
- GAME
- GCF
- CONVERT

Metodat shtese te implementuara:

- LOTTO
- HIDE

2. Hyrja

2.1 Shpjegim I shkurter per projektin

Qellimi I ketij projekti ishte realizimi I komunikimit ndermjet serverit dhe klientit I cili eshte bere permes socket-ave.

FIEK Protokoli eshte nje protokol I thjeshte ku komunikimi ndermjet klientit dhe serverit ka per baze llojin e kerkeses(apo fjaline) e shenuar nga klienti. Serveri I kthen pergjigje te caktuar klientit varesisht nga kerkesa e tij e kjo behet duke krijuar metodat per keto kerkesa. Komunikimi eshte realizuar permes dy protokoleve TCP dhe UDP.

Kerkesat nga te cilat klienti mund te pres pergjigje jane : IP Adresa, Numri Portit, Loja, Koha , Numri Bashketingelloreve dhe zanore , Te tregoj se nje fjali eshte palindrome apo jo, Konvertimi I njesive, GCF faktori, Nje loje llotarie si dhe fshahja e caktuar e disa karaktereve te tekstit per arsye sigurie(si metoda shtese).

3. Pershkrim I kodit te aplikacionit te serverit

3.1 Pershkrimi I serverit TCP

Serveri lidhet me klient permes socketave dhe per secilen lidhje me klient ai krijon nje thread te ri , pasi qe per cdo transport qe duhet bere nga server ne klient krijohet(nevojitet) nje thread i ri. Gjate tere kohes server eshte aktiv per te pritur kerkesa nga klienti.

- Fillimisht eshte bere importimi i librarive te gatshme.

```
from numpy import gcd
from _thread import start_new_thread
import socket
import random
import datetime
```

- Krijojme variablen TCP_Server dhe TCP_Port qe mundeson komunikim me client ne IP dhe portin e caktuar!
- Krijohet variabla serverSocket qe mundeson pranimin e paremtrave (familjen e ip_adresave dhe menyren e komunikimit qe ne kete rast eshte tcp)
- Listen(1) mundeson qe nje perdoruesi ti rezervohet socketi

```
TCP_Server = "localhost"
TCP_Port = 13000

serverSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
serverSocket.bind((TCP_Server, TCP_Port))
serverSocket.listen(1)
```

- Krijohet Funkcioni newThread qe mundeson marrjen e shenimeve nga klienti
- Madhesia maksimale e te dhenave duhet te jete 128 bita!
- Largojme hapesirat nga fillimi ne fund me strip() dhe perdorim split() qe shikojme cfare ka shenuar klienti ma ane te variables socketReceive dhe request
- Ne qofte se ka ndonje gabim lidhja do te shkeputet por serveri nuk mbyllet

```
def newThread(socket, clAddress):
    try:
        Receive = socket.recv(128)
        socketReceive = str(Receive.decode().strip())
        request = Receive.decode("utf-8").split(' ')
```

```
except:
    return
```

- Unaza While eshte krijuar per te pranuar kerkesa dhe eshte unaza e pafundme e cila mundeson qe serveri te pranon kerkesa tere kohen derisa eshte funksional!
- Lidhja me klientin pranohet nqs adresohet permes portit 13000

```
while True:
    try:
        socket, clAddress = serverSocket.accept()
        start_new_thread(newThread, (socket, clAddress))
    except:
        continue
serverSocket.close()
quit()
```

- Thirrja e metodave mundesohet permes argumenteve

```

try:
    if socketReceive == "IPADDRESS":
        socket.send(IPADDRESS().encode())
    elif socketReceive == "PORT":
        socket.send(PORT().encode())
    elif request[0].upper() == "COUNT":
        COUNT(" ".join(request[1:]), socket)
    elif request[0].upper() == "REVERSE":
        REVERSE(" ".join(request[1:]), socket)
    elif request[0].upper() == "PALINDROME":
        PALINDROME(" ".join(request[1:]), socket)
    elif socketReceive == "TIME":
        socket.send(TIME().encode())
    elif socketReceive == "GAME":
        socket.send(GAME().encode())
    elif request[0].upper() == "GCF":
        try:
            GCF(int(request[1]), int(request[2]), socket)
        except Exception:
            raise Exception("GCF Argumentet Gabim/Mungojne!")
    elif request[0].upper() == "CONVERT":
        try:
            CONVERT(request[1], float(request[2]), socket)
        except Exception:
            raise Exception("Convert Argumentet Gabim/Mungojne!")
    elif request[0].upper() == "LOTTO":
        try:
            LOTTO(int(request[1]), int(request[2]), int(request[3]), socket)
        except Exception:
            raise Exception("LOTTO Argumentet Gabim/Mungojne!")
    elif request[0].upper() == "HIDE":
        HIDE(" ".join(request[1:]), socket)

```

- Funksioni I mesiperm gjithashtu trajton edhe gabimet ne argumente si me poshte

```

except Exception as error:
    var = ""

    if str(error) == "GCF Argumentet Gabim/Mungojne!":
        var = "\nGCF\ pranon 2 numra integjer si argumente!"
    elif str(error) == "Convert Argumentet Gabim/Mungojne!":
        var = "\nCONVERT\ args must be (cmToFeet, FeetToCm, kmToMiles, MilesToKm) and its value to convert!"
    elif str(error) == "LOTTO Argumentet Gabim/Mungojne!":
        var = "\nLOTTO\ Miresevini ne Lotto. Zgjedh tre numra prej rangut 0-20"

    socket.send(var.encode("utf-8"))
    socket.close()

```

3.1 Pershkrimi I serverit UDP

Serveri UDP mundeson lidhjen me servere dhe komunikimin permes Datagrameve

- Fillimisht bejme importimin e librarive te gatshme
- Definojme portin e serverit I cilli mundeson komunikim me klient permes porteve te njejta

- Krijimi I variables serverSocket qe mundeson pranimin e paremetrave qe tregojne familjen e IP-adresat dhe menyren e komunikimit (UDP ne kete rat)
- Permes komandes bind serverit I rezervohet porti I klientit

```
from numpy import gcd
import socket
import random
import datetime

UDP_Server = "localhost"
UDP_Port = 13000

serverSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
serverSocket.bind((UDP_Server, UDP_Port))
```

- Krijojme nje unaze te pafundme qe mundeson serveri te pranoj kerlesa tere kohen
- Krijojme varablat socket dhe request qe mundesojne pranimin e kerkeses se klientit ku gjatesia e karakterit duhet te jete 128 biteshe permes enkodimit I cili mundeson edhe largimin e hapesirava ndermjet tekstit
- Dekodojme kerkesen e marre ng klienti te cilen e kthejme ne string ku te gjitha karakteret kthehen si upercase dhe largohen hapesirat ne fillim dhe ne fund

```
while True:
    socket, clAddress = serverSocket.recvfrom(128)
    request = socket.decode("utf-8").split(" ")
    socketReceive = str(socket.decode().strip().upper())
```

- Thirrja e argumenteve behet ne menyre te ngjeshme edhe tek UDP-Serveri vecse ne rast gabimi serveri nuk do te mbyllet por paralajmeron per kerkesa te sakta te metodava!

```
else:
    serverSocket.sendto("Pass the method right! Type \"QUIT\" if you want to quit".encode("utf-8"),
                       clAddress)
```

4.Pershkrimi I TCP-Klientit

- Se pari deklarojme variablat TCP_Server dhe TCP_Port
- Pastaj Klienti tentin lidhjen me serverin permes unazes while
- Nese Serveri eshte i hapur krijojme socketin e klientit clSocket I cili pranon familjen e IPAdresave dhe menyrene komunikimit
- Deklarohet varabla input e cila mundeson pranimin e argumenteve nga klienti si string dhe I dergon ne server me gjatesi 128 byte
- Nqs kerkesa e klientit eshte QUIT klienti shkeputet nga serveri!
- Nese Serveri nuk eshte aktiv Klienti pranon nje tekst qe tregon qe Serveri nuk eshte aktiv dhe mbyllet unaza

```

import socket

TCP_Server = "localhost"
TCP_Port = 13800

while True:
    try:
        clSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        clSocket.connect((TCP_Server, TCP_Port))

        print("Operationi (IPADDRESS, PORT, COUNT, REVERSE, PALINDROME, TIME, GAME, GCF, CONVERT, LOTTO, HIDE)? ", end="...")
        request = input("")

        if request == "":
            continue
        elif request == "QUIT":
            print("Ju u shkyqet nga serveri!\n")
            break

        clSocket.send(request.encode("utf-8"))

        NewData = clSocket.recv(128)

        print(NewData.decode("utf-8") + "\n")

    except:
        print("Serveri nuk eshte aktiv!")
        break

    clSocket.close()
    quit()

```

4.Pershkrimi I UDP- Klientit

- Se pari deklarojme variablat UDP_Server dhe UDP_Port qe rezervon portin per lidhje me serverin
- Pastaj klienti rezervon nje vend edhe ne rast se serveri nuk eshte aktiv permes funksionit socket qe prannon dy parametra te lartcekur tek UDP Server
- Ne rasrt se serveri eshte aktiv atehere inputi pranohet si string nga Serveri dhe kthen kerkesen por nuk mbyll lidhjen
- Kerkesa pranohet permes variablave Data e cila duhet te jete e gjate 128 byte
- Ne rast se serveri nuk eshte aktiv klientit do ti shfaqet nje mesazh qe tregon qe serveri nuk eshte aktiv dhe permes komandes QUIT mundesohet shkyqja e klientit


```

import socket

UDP_Server = "localhost"
UDP_port = 13000

clSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

while True:
    try:
        print("Operacioni(IPADDRESS, PORT, COUNT, REVERSE, PALINDROME, TIME, GAME, GCF, CONVERT, LOTTO, HIDE)?", end=" ")
        request = input("")

        if request == "":
            continue
        elif request.upper() == "QUIT":
            print("Ju u shkyget nga serveri!\n")
            break

        clSocket.sendto(request.encode("utf-8"), (UDP_Server, UDP_port))

        Data, address = clSocket.recvfrom(128)

        print(Data.decode("utf-8"))
    except:
        print("Serveri nuk eshte aktiv (Shtypni QUIT per tu larguar).\n")

clSocket.close()
quit()

```

5. Pershkrimi I metodave

- IPADDRESS

Pergjiegja: Pecakton dhe kthen IP adresen e klientit

```

def IPADDRESS():
    return ('IP Adresa e klientit e klientit eshte: %s ' % clAddress[0])

```

- PORT

Pergjigja: Percakton dhe kthen portin e klientit(jo te serverit).

```

def PORT():
    if clAddress[1] is not None:
        return 'Klienti është duke përdorur portin: %s ' % clAddress[1]

```

- COUNT

Pergjiegja: Numrin numrin e zanoreve dhe bashktingelloreve te nje teksti me karaktere te gjuhes angleze te pranuar nga klienti me pas dergon ate tek klienti

```
def COUNT(text, socket):
    if text != "":
        zanore = 0
        bashktingellore = 0
        for char in text:
            if char in "aeiouyAEIOUY":
                zanore += 1
            elif char in "bcdfghiklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ":
                bashktingellore += 1
        var = "Teksti i pranuar përmban " + str(zanore) + " zanore dhe " + str(bashktingellore) + " bashktingellore"
        socket.send(var.encode("utf-8"))
    else:
        socket.send("NULL!".encode("utf-8"))
```

- REVERSE

Pergjiegja: Pranon nje tekst nga klienti dhe dergon ate te kthyer mbrapsht tek klienti. Se pari largohen hapesirat ne fillim dhe fund

```
def REVERSE(text, socket):
    revtext = ""
    for char in text:
        revtext = char + revtext
    var = revtext.strip()
    socket.send(var.encode("utf-8"))
```

- PALINDROME

Pergjiegje: Pranon nje tekst nga klienti dhe me pas dergon nje bool type tek klienti qe tregon se fjalia e dhene eshte palindrome apo jo

```
def PALINDROME(text, socket):
    if text == text[::-1]:
        var = str(True)
    else:
        var = str(False)
    socket.send(var.encode("utf-8"))
```

- TIME

Pergjiegja: Kthen kohen aktuale tek klienti mundesohet permes librarise datetime

```
def TIME():
    if datetime.datetime.now() is not None:
        return datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
```

- GAME

Pergjiegja: Kthen nje list qe permban 5 numra random te rangut 1,35

```
def GAME():
    var = [random.randint(1, 35) for rand in range(0, 5)]
    var.sort()
    return str(var)
```

- GCF

Pergjiegja: Tregon faktorin me te vogel perbashket te numrave te dhene nga klienti dhe kthen rezultatin tek klienti

```
def GCF(number1, number2, socket):
    var = str(gcd(number1, number2))
    socket.send(var.encode("utf-8"))
```

- CONVERT

Pergjiegja: Mundeson konvertimin e njesive te caktuara te cilat mun te therriren nga klienti permes metodave te caktuara

```
def CONVERT(option, number, socket):
    var = None
    if option == "cmToFeet":
        var = number * 0.0328084
    elif option == "FeetToCm":
        var = number * 30.48
    elif option == "kmToMiles":
        var = number * 0.621371
    elif option == "MilesToKm":
        var = number * 1.609344
    else:
        raise Exception("wrong convert option")
    var = str(var)
    socket.send(var.encode("utf-8"))
```

- LOTTO(Metoda 1 Shtese)

Pergjiegje: Se pari klienti duhet te zgjedh tre numra jo te perseritur te rangut 0-20. Serveri gjeneron gjithashtu tre numra te randomte jo te perseritur te atij rangui. Me pas Serveri dergon informata klientit se cilat numra ka perzgjedhur sakte, dhe sa te tille. Numrat nuk kane nevojte te jene ne renditje te njejte

```
# Metoda ---1#
def LOTTO(number1, number2, number3, socket):
    numbers = [number1, number2, number3]
    nr_zgjedhur = numbers
    generated = random.sample(range(21), 3)
    nr_sakte = []
    for numbers in nr_zgjedhur:
        if numbers in generated:
            nr_sakte.append(numbers)

    var = "Nr qe ju zgjedhet: " + str(nr_zgjedhur)
    var1 = "\nNr e gjeneruar: " + str(generated)
    var2 = "\nNr qe ju qelluat: " + str(nr_sakte)
    count_nr_sakte = len(nr_sakte)
    var3 = "\nJu keni qelluar: " + str(count_nr_sakte) + " numra!"
    var4 = "\nTRY AGAIN!"
    if count_nr_sakte == len(generated):
        var4 = "\nWIN!"
    socket.send(
        var.encode("utf-8") + var1.encode("utf-8") + var2.encode("utf-8") + var3.encode(
            "utf-8") + var4.encode("utf-8"))
```

- HIDE(Metoda 2 Shtese)

Pergjiegje: Se pari klienti dergon tek serveri nje tekst te qfaredoshem. Me pas Serveri pranon mesazhin dhe kthen ate te fshehur perpos 4 karaktereve te fundit

```
# Metoda ---2#
def HIDE(text, clAddress):
    var = '*' * (len(text) - 4) + text[-4:]
    serverSocket.sendto(var.encode("utf-8"), clAddress)
```

6.REZULTATET E PRANUARA NGA THIRRJA E METODAVE

```
Operacioni(IPADDRESS, PORT, COUNT, REVERSE, PALINDROME, TIME, GAME, GCF, CONVERT, LOTTO, HIDE)? IPADDRESS
IP Adresa e klientit është: 127.0.0.1
Operacioni(IPADDRESS, PORT, COUNT, REVERSE, PALINDROME, TIME, GAME, GCF, CONVERT, LOTTO, HIDE)? PORT
Klienti është duke përdorur portin: 41388
Operacioni(IPADDRESS, PORT, COUNT, REVERSE, PALINDROME, TIME, GAME, GCF, CONVERT, LOTTO, HIDE)? COUNT      LI RIM
Teksti i pranuar përmban 2 zanore dhe 3 bashketingellore
Operacioni(IPADDRESS, PORT, COUNT, REVERSE, PALINDROME, TIME, GAME, GCF, CONVERT, LOTTO, HIDE)? REVERSE      MI RIL
LIR IM
Operacioni(IPADDRESS, PORT, COUNT, REVERSE, PALINDROME, TIME, GAME, GCF, CONVERT, LOTTO, HIDE)? PALINDROME AHA
True
Operacioni(IPADDRESS, PORT, COUNT, REVERSE, PALINDROME, TIME, GAME, GCF, CONVERT, LOTTO, HIDE)? PALINFROME LIRIM
Pass the method right! Type "QUIT" if you want to quit
Operacioni(IPADDRESS, PORT, COUNT, REVERSE, PALINDROME, TIME, GAME, GCF, CONVERT, LOTTO, HIDE)? PALINDROME LIRIM
False
Operacioni(IPADDRESS, PORT, COUNT, REVERSE, PALINDROME, TIME, GAME, GCF, CONVERT, LOTTO, HIDE)? TIME
2020-04-10 22:40:05
Operacioni(IPADDRESS, PORT, COUNT, REVERSE, PALINDROME, TIME, GAME, GCF, CONVERT, LOTTO, HIDE)? GAME
[1, 2, 5, 9, 34]
Operacioni(IPADDRESS, PORT, COUNT, REVERSE, PALINDROME, TIME, GAME, GCF, CONVERT, LOTTO, HIDE)? CONVERT cmToFeet 100
3.28084
Operacioni(IPADDRESS, PORT, COUNT, REVERSE, PALINDROME, TIME, GAME, GCF, CONVERT, LOTTO, HIDE)? LOTTO 1 10 20
Nr qe ju zgjedhet: [1, 10, 20]
Nr e gjeneruar: [7, 19, 9]
Nr qe ju qelluat: []
Ju keni qelluar: 0 numra!
TRY AGAIN!
Operacioni(IPADDRESS, PORT, COUNT, REVERSE, PALINDROME, TIME, GAME, GCF, CONVERT, LOTTO, HIDE)? LOTTO 1 2 3
Nr qe ju zgjedhet: [1, 2, 3]
Nr e gjeneruar: [6, 5, 16]
Nr qe ju qelluat: []
Ju keni qelluar: 0 numra!
TRY AGAIN!
Operacioni(IPADDRESS, PORT, COUNT, REVERSE, PALINDROME, TIME, GAME, GCF, CONVERT, LOTTO, HIDE)? HIDE 1234 1234 1234 1234
***** 1234
Operacioni(IPADDRESS, PORT, COUNT, REVERSE, PALINDROME, TIME, GAME, GCF, CONVERT, LOTTO, HIDE)? QUIT
Ju u shkyqet nga serveri!
```