**ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**"ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"**

Лабораторная работа № 4

Тема: «Разработка простейших клиент-серверных программ с использованием TCP сокетов на базе ОС Linux.»

Проверил: Выполнил:

асс. каф. ПИ ст. гр. ПИ-18б

Ищенко А.П. Моргунов А.Г.

\_\_\_\_.\_\_\_\_.2021г. \_\_\_\_.\_\_\_\_.2021г.

ст. преп. ПИ

Чернышова А.В.

\_\_\_\_.\_\_\_\_.2021г.

Донецк – 2021

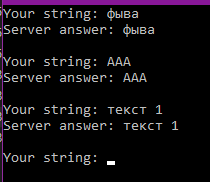
**Задание**

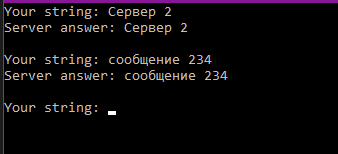
Откомпилируйте программы простейших TCP-клиента и ТСР-сервера (последовательная обработка запросов TCP-сервером). Убедитесь в правильном функционировании этих программ.

Изучите системные вызовы, которые используются в серверной и клиентской частях.

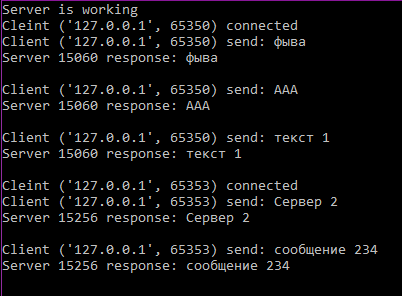
Напишите программу взаимодействия клиента и сервера на базе TCP соединения, используя параллельную обработку запросов TCP-сервером, ограничьте количество соединений (не более указанного числа) и время ожидания сервером запроса от клиента при уже установленном соединении.

**Скриншоты работы TCP-клиентов**





**Скриншоты работы TCP-сервера**



**Листинг программы**

Клиент

import socket

client\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

client\_socket.settimeout(2)

try:

client\_socket.connect(("localhost", 51000))

except Exception:

print("Connection error")

while True:

message = input("Your string: ")

if message == "exit":

try:

client\_socket.send(message.encode())

except Exception:

pass

client\_socket.close()

print("Exit")

break

message = message.encode()

try:

client\_socket.send(message)

message = client\_socket.recv(4096)

except Exception:

print("Server error")

print("Exit")

client\_socket.close()

break

message = message.decode()

print("Server answer: {}\n".format(message))

Сервер

import socket

import threading

num = 0

def client(client\_socket, client\_address):

global num

print("Cleint {} connected".format(client\_address))

while True:

try:

message = client\_socket.recv(4096)

except Exception:

print("Client {} disabled by timeout\n".format(client\_address))

num-=1

print(num)

break

message = message.decode()

if message == "exit":

client\_socket.close()

print("Client {} was closed".format(client\_address))

num-=1

print(num)

break

print("Client {} send: {}".format(client\_address, message))

print("Server {} response: {}\n".format(

threading.get\_ident(), message))

client\_socket.send(message.encode())

print("Server is working")

server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server\_socket.bind(("localhost", 51000))

server\_socket.listen(5)

while True:

socket, address = server\_socket.accept()

num+=1

if (num>3):

socket.close()

num-=1

continue

socket.settimeout(60)

threading.Thread(target=client, args=(socket, address)).start()