Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

Лабораторная работа №3

по дисциплине "Информационные сети"

ТЕМА РАБОТЫ:

ВЗАИМОДЕЙСТВИЕ ПРИКЛАДНЫХ ПРОГРАММ С ПОМОЩЬЮ ТРАНСПОРТНЫХ ПРОТОКОЛОВ СЕТИ ИНТЕРНЕТ

Выполнил:

студент гр. ПРИ-120

Парахин К.В.

Принял:

Доцент кафедры ИСПИ

Курочкин С.В.

Владимир 2022 г.

Цель работы:

Изучение принципов работы транспортных протоколов Интернет, разработка прикладных программ, осуществляющих взаимодействие между собой на основе этих протоколов

Выполнение задания:

Организовать взаимодействие типа клиент - сервер. К серверу одновременно может подключиться только один клиент. Остальные клиенты заносятся в очередь, и им высылается сообщение об ожидании освобождения сервера (Вариант 24)

Листинг программ:

Программа server.py на Apache веб-сервере:

import socket

sock = socket.socket()

sock.bind(('', 5000))

sock.listen(10)

print('Server is running...')

clientsQueue = {}

def checkPossibiltyToConnect(conn, adr):

data = conn.recv(1024)

if !(addr in clientsQueue):

clientsQueue[addr[0]] = {'User' : data}

conn.send(bytes('connection\_isAwaited, encoding = 'UTF-8')

return False

else:

conn.send(bytes('connection\_OK', encoding = 'UTF-8')

return True

while True:

conn, addr = sock.accept()

if checkPossibiltyToConnect(conn, adr):

print('Connected: ', addr)

data = conn.recv(1024)

print(str(data))

conn.send(data)

conn.close()

Программа client.py, исполняемая на клиенте:

import socket

sock = socket.socket()

ip = input("Введите свой IP-адрес: \n")

port = input("Введите свой порт: \n")

sock.bind((ip, int(port)))

ipServer = input("Введите IP-адрес сервера: \n")

portServer = input("Введите порт сервера: ")

def tryConnect():

data = sock.recv(1024)

serverMsg = data.decode('UTF-8')

if (serverMsg == 'connection\_OK'):

return True

elif (serverMsg == 'connection\_isAwaited'):

return False

sock.connect((ipServer, int(portServer)))

if (tryConnect()):

print("Подключение к серверу успешно прошло: ")

msg = bytes(input("Введите сообщение: \n"), encoding = 'UTF-8')

sock.send(msg)

data = sock.recv(1024)

print(str(data))

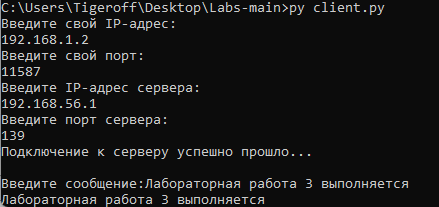
else:

print("Сервер в настоящий момент занят, пожалуйста, подождите…")

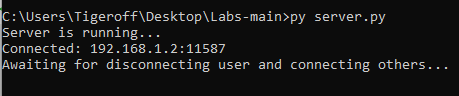
sock.close()

Тестирование работы программ:

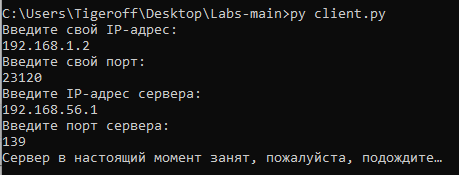
В окне консоли запускаем клиентский скрипт и пробуем войти на пустой сервер:



В это время в отдельном окне консоли выполняется скрипт сервера:



В это время запустим в новом окне консоли снова клиентский скрипт и попробуем зайти туда новым пользователем, пока сервер занят старым пользователем:



Вывод

В результате выполнения работы, я изучил принципы работы транспортных протоколов Интернет, а также познакомился с разработкой прикладных программ, осуществляющих взаимодействие между собой на основе этих протоколов