УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе № 3

по предмету «Компьютерные системы и сети»

Выполнил:

Горник А.

гр. 351003

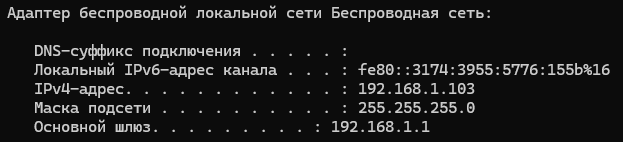
Проверил:

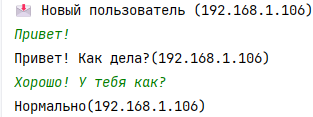
Леванцевич В.А.

Минск, 2025

**Код программы на python**  
import socket  
import threading  
import time  
  
PORT = 5000  
TCP\_PORT = 6000  
BUFFER\_SIZE = 1024  
my\_ip = "192.168.1.102"  
user\_list = set() *# Уникальные пользователи*continue\_signal = True  
  
*# Функция для приема сообщений (UDP сервер)*def receive\_messages():  
 global continue\_signal  
 *# Создаем UDP сокет (IPv4, UDP)* server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)  
 *# Разрешаем повторное использование адреса* server\_socket.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1)  
 *# Привязываем сокет к IP и порту* server\_socket.bind((my\_ip, PORT))  
 while continue\_signal:  
 try:  
 *# Получаем только адрес отправителя* \_, addr = server\_socket.recvfrom(BUFFER\_SIZE)  
 user\_ip = addr[0]  
 if user\_ip != my\_ip and user\_ip not in user\_list:  
 print(f"📩 Новый пользователь ({user\_ip})")  
 user\_list.add(user\_ip)  
 except:  
 break  
  
*# Функция для отправки сообщений и поиска новых узлов (UDP клиент)*def send\_broadcast():  
 global continue\_signal  
 *# Создаем UDP клиентский сокет (IPv4, UDP)* client\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)  
 *# Включаем возможность отправки широковещательных сообщений всем в подсети* client\_socket.setsockopt(socket.SOL\_SOCKET, socket.SO\_BROADCAST, 1)  
 broadcast\_address = (f"{my\_ip[:my\_ip.rfind('.')]+'.255'}", PORT)  
 while continue\_signal:  
 try:  
 client\_socket.sendto("".encode(), broadcast\_address)  
 time.sleep(5)  
 except:  
 break  
 client\_socket.close()  
  
*# Функция для установки TCP соединений с каждым узлом*def establish\_tcp\_connections():  
 global continue\_signal  
 while continue\_signal:  
 for user\_ip in list(user\_list):  
 try:  
 *# Создаем TCP сокет (IPv4, TCP)* client\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
 *# Подключаемся к пользователю по IP и TCP порту* client\_socket.connect((user\_ip, TCP\_PORT))  
 message = input()  
 if message == "0":  
 continue\_signal = False  
 message = "Выхожу из чата"  
 *# Отправляем сообщение пользователю* client\_socket.sendall(message.encode())  
 client\_socket.close() *# Закрываем соединение* except:  
 break  
 time.sleep(10)  
  
*# Функция для приема входящих TCP соединений (TCP сервер)*def tcp\_server():  
 global continue\_signal  
 *# Создаем TCP серверный сокет (IPv4, TCP)* server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
 *# Привязываем сокет к IP и порту для прослушивания входящих подключений* server\_socket.bind((my\_ip, TCP\_PORT))  
 while continue\_signal:  
 try:  
 server\_socket.listen(1)  
 conn, addr = server\_socket.accept()  
 data = conn.recv(BUFFER\_SIZE).decode()  
 print(f"{data}({addr})")  
 conn.close()  
 except:  
 break  
  
*# Запускаем все параллельно*receive\_thread = threading.Thread(target=receive\_messages)  
send\_thread = threading.Thread(target=send\_broadcast)  
tcp\_server\_thread = threading.Thread(target=tcp\_server)  
tcp\_client\_thread = threading.Thread(target=establish\_tcp\_connections)  
  
receive\_thread.start()  
send\_thread.start()  
tcp\_server\_thread.start()  
tcp\_client\_thread.start()  
  
*# Ждем завершения всех потоков*receive\_thread.join()  
send\_thread.join()  
tcp\_server\_thread.join()  
tcp\_client\_thread.join()  
  
print("🔚 Завершение работы!")  
  
*#Метод setsockopt() с параметрами socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1 выполняет следующие функции:  
#Уровень сокета (socket.SOL\_SOCKET): Это означает, что мы устанавливаем опцию на уровне самого сокета, а не какого-либо протокола (например, TCP или UDP).  
#Опция SO\_REUSEADDR: Эта опция позволяет сокету использовать локальный адрес (IP и порт), даже если он недавно использовался другим сокетом, который мог быть  
#закрыт не полностью или находится в состоянии TIME\_WAIT. Без этой опции операционная система может блокировать повторное использование того же адреса, что  
#приведет к ошибке «Address already in use», если вы попытаетесь быстро перезапустить сервер.  
#Значение 1: Устанавливая значение равное 1 (что соответствует True), мы включаем эту опцию.*

**Скриншоты работы программы**

****

****