

Министерство Образования, Культуры и Исследований
Технический Университет Молдовы Департамент Программной
Инженерии и Автоматики

ОТЧЕТ

Лабораторная работа №5

По предмету: **Программирование в сети**

Тема: Aplicație Client-Server TCP.

Выполнил студент гр.SI-212:

Решетников Максим

Проверил:

Лях Аркадий

Кишинев 2024 г.

Ссылка на [GitHub](#)

Клиент

```
import socket
import tkinter as tk

def send_message():
    message = entry_message.get()
    if message:
        try:
            client_socket.send(message.encode('utf-8'))
            response = client_socket.recv(1024)
            print("Ответ от сервера:", response.decode('utf-8'))
        except Exception as e:
            print("Ошибка при отправке сообщения:", e)
    else:
        print("Введите сообщение для отправки")

root = tk.Tk()
root.title("Клиент TCP")

label_message = tk.Label(root, text="Сообщение:")
label_message.grid(row=0, column=0, padx=5, pady=5)

entry_message = tk.Entry(root, width=40)
entry_message.grid(row=0, column=1, padx=5, pady=5)

button_send = tk.Button(root, text="Отправить", command=send_message)
button_send.grid(row=1, column=0, columnspan=2, padx=5, pady=5, sticky="we")

SERVER_HOST = '127.0.0.1'
SERVER_PORT = 12345

client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
try:
    client_socket.connect((SERVER_HOST, SERVER_PORT))
except Exception as e:
    print("Не удалось подключиться к серверу:", e)
    root.destroy()
else:
    root.mainloop()

client_socket.close()
```

Сервер

```
import socket
import threading

# Функция, которая будет выполняться для каждого клиента
def handle_client(client_socket, address):
    print(f"Подключение от {address}")

    # Получение и отправка данных
    while True:
        data = client_socket.recv(1024)
        if not data:
            break
        response = f"Получено сообщение: {data.decode('utf-8')}"
        client_socket.send(response.encode('utf-8'))

    # Закрытие соединения с клиентом
    print(f"Соединение с {address} закрыто")
    client_socket.close()

# Адрес и порт, на которых сервер будет прослушивать подключения
SERVER_HOST = '127.0.0.1'
SERVER_PORT = 12345

# Создание сокета для сервера
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind((SERVER_HOST, SERVER_PORT))
server_socket.listen(5)
print(f"Сервер слушает на {SERVER_HOST}:{SERVER_PORT}")

while True:
    client_socket, address = server_socket.accept()
    client_thread = threading.Thread(target=handle_client, args=(client_socket,
address))
    client_thread.start()
```

Вывод: В ходе работы мы научились создавать Client-Server TCP приложение.