



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

Институт № 3 «Системы управления, информатика и электроэнергетика»

Кафедра 304 «Вычислительные машины, системы и сети»

Лабораторная работа № 2
по дисциплине «WEB технологии»
на тему «Сокеты»

Выполнил
студент группы МЗО-125БВ-24
Егоров А.В.

Приняли
ассистент каф. 304 Борисов А.И.

Москва
2025

Содержание:

Постановка задачи.....	3
Код синхронного сервера (python).....	5
Код асинхронного сервера (python).....	7
Код клиента (node js).....	9
Конфигурационные файлы программ.....	11
Логи.....	12
Описание работы программы.....	16
Вывод по работе:.....	19
Ссылки.....	19

Постановка задачи:

1. Разработать программу-клиент, работающую на сокетах, на выбранном Вами языке программирования. Данный клиент должен хранить файл логов соединений.

В логах хранится время подключения и адрес сервера, время отправки сообщения и само сообщение, время получения сообщения и само сообщение. При подключении к серверу клиент через определенный промежуток времени (задается разработчиком клиента) отправляет сообщение серверу: ФИО и группа студента, а также ждет от сервера. Данные о сервере необходимо вводить через файл конфигурации. В отчете привести код клиента и один тестовый лог файл.

2. Разработать программу-сервер, которая ожидает строку от клиента. Через промежуток времени (моделирование работы сервера) он отправляет в ответ эту же строку, только зеркально отраженную и добавляет к сообщению ФИО студента, написавшего сервер. (пример ответа: Чиволвап Йирю Вотит. Сервер написан Титовым Ю.П. МЗО-1ХХБВ-ХХ). Сервер должен эмулировать работу путем временной задержки. Отключение клиентов от сервера совершать через определенное время работы клиента.

Сервер и клиент должны быть написаны на различных языках программирования.

Сервер ведёт лог-файл, в котором содержится:

- Время запуска сервера,
- Время подсоединения клиента,
- Адрес клиента,
- Время получения сообщения и само сообщение,
- Время отправки сообщения и само сообщение, Время отключения клиента от сервера.

Лог-файл не очищать.

3. Доработать программу-сервер для выполнения асинхронных запросов от клиентов. Чётные варианты реализуют многопоточный асинхронный сервер, нечетные – многопроцессный. Номер варианта определяется номером по списку.

При доработке сервера необходимо создать новый проект, т.е. в результате должно получиться две программы-сервера.

4. Запустить в сети кафедры 5 клиентов и 1 сервер. В отчете привести настройки для подключения и логи клиентов и серверов. Проводить тестирование одновременного подключения, т.е. второй, третий и т.д. пользователь подключается к серверу тогда, когда первый пользователь еще не получил ответ.

Желательно провести тестирование как на однопоточном, так и на асинхронном сервере, описать отличия.

Код синхронного сервера (python)

```
import socket, time, json, os, logging

# Создание логгера
logger = logging.getLogger("logger")
logger.setLevel(logging.INFO) # Уровень логирования

# Формат логов
formatter = logging.Formatter(
    "%(asctime)s - %(levelname)s - %(message)s",
    datefmt="%Y-%m-%d %H:%M:%S"
)

# Логгер для терминала
console_logger = logging.StreamHandler()
console_logger.setFormatter(formatter)
logger.addHandler(console_logger)

# Логгер для файла
file_log = logging.FileHandler("logs.log", mode="a",
encoding="utf-8")
file_log.setFormatter(formatter)
logger.addHandler(file_log)

JSON_FILE = "config.json"

# Чтение конфига из файла
if os.path.exists(JSON_FILE):
    with open(JSON_FILE) as file:
        try:
            config = json.load(file)
        except Exception as e:
            logger.error(f"Ошибка: произошла ошибка при чтении
конфигурационного файла: {e}. Выход.")
            exit()
else:
    logger.error("Ошибка: конфигурационный файл не найден. Выход.")
    exit()

# Запись данных из файла в переменную
port = int(config.get("port"))
ip_addres = str(config.get("ip_addres"))
handler_delay = int(config.get("handler_delay"))
working_time = int(config.get("working_time"))

# Создание сокета
sock = socket.socket()
sock.bind((ip_addres, port))
sock.listen(1)
```

```

logger.info("Сервер запущен. Ожидание сообщений...")

def main():
    # Получение данных о подключенном клиенте
    conn, addr = sock.accept()
    conn.settimeout(1) # тайм-аут сокета
    logger.info(f"Установлено соединение с клиентом: {addr}")

    working_end_time = time.time() + working_time

    while time.time() < working_end_time:
        # Получение данных от клиента
        try:
            data = conn.recv(1024)
            data = data.decode()

            logger.info(f"Получено сообщение от клиента {addr}:
'{data}')"

            # Эмуляция работы сервера
            time.sleep(handler_delay)

            # Отправка "обработанных" данных клиенту
            message = f"{data[:: -1]}. Сервер написан Егоровым А.В.
М30-125БВ-24"
            conn.send((message).encode())

            logger.info(f"Сообщение отправлено клиенту {addr}:
'{message}')"
        except socket.timeout:
            continue

        # Закрытие соединения
        # conn.send(("Клиент отключен от сервера.")).encode())
        conn.close()

        logger.info(f"Клиент отключен от сервера: {addr}")

while True:
    main()

```

Код асинхронного сервера (python)

```
import socket, time, json, os, logging, asyncio

# Логгер
logger = logging.getLogger("logger")
logger.setLevel(logging.INFO)
formatter = logging.Formatter("%(asctime)s - %(levelname)s - %(message)s",
datefmt="%Y-%m-%d %H:%M:%S")

console_logger = logging.StreamHandler()
console_logger.setFormatter(formatter)
logger.addHandler(console_logger)

file_log = logging.FileHandler("logs.log", mode="a", encoding="utf-8")
file_log.setFormatter(formatter)
logger.addHandler(file_log)

# Чтение конфигурации
JSON_FILE = "config.json"
if os.path.exists(JSON_FILE):
    with open(JSON_FILE) as file:
        try:
            config = json.load(file)
        except Exception as e:
            logger.error(f"Ошибка: произошла ошибка при чтении
конфигурационного файла: {e}. Выход.")
            exit()
else:
    logger.error("Ошибка: конфигурационный файл не найден. Выход.")
    exit()

# Настройки
port = int(config.get("port"))
ip_address = str(config.get("ip_address"))
handler_delay = int(config.get("handler_delay"))
working_time = int(config.get("working_time"))

# Создание сокета
sock = socket.socket()
sock.bind((ip_address, port))
sock.listen()
sock.setblocking(False)

logger.info("Сервер запущен. Ожидание сообщений...")

async def handle_client(conn, addr):
    logger.info(f"Установлено соединение с клиентом: {addr}")
    conn.setblocking(False)

    loop = asyncio.get_running_loop()
    working_end_time = time.time() + working_time

    while time.time() < working_end_time:
        try:
            data = await asyncio.wait_for(loop.sock_recv(conn, 1024),
timeout=1.0)
```

```

        data = data.decode()
        logger.info(f"Получено сообщение от клиента {addr}: '{data}'")

        await asyncio.sleep(handler_delay)

        message = f"{data[:: -1]}. Сервер написан Егоровым А.В.
М30-125БВ-24"
        await loop.sock_sendall(conn, message.encode())
        logger.info(f"Сообщение отправлено клиенту {addr}: '{message}'")
    except asyncio.TimeoutError:
        continue
    except ConnectionResetError:
        logger.warning(f"Клиент {addr} разорвал соединение.")
        break
    except Exception as e:
        logger.error(f"Ошибка при обработке клиента {addr}: {e}")
        break

    conn.close()
    logger.info(f"Клиент отключен от сервера: {addr}")

async def main():
    loop = asyncio.get_running_loop()

    while True:
        conn, addr = await loop.sock_accept(sock)
        asyncio.create_task(handle_client(conn, addr))

# Запуск
if __name__ == "__main__":
    try:
        asyncio.run(main())
    except KeyboardInterrupt:
        logger.info("Сервер остановлен вручную.")

```


Код клиента (node js)

```
const net = require("net");
const fs = require("fs");
const os = require("os");

const CONFIG_FILE = "config.json";
const LOG_FILE = "logs.log";
const MESSAGE = "Егоров Александр Владиславович М30-125БВ-24";

// Логирование
function log(message, level = "INFO") {
  const timestamp = new Date().toISOString().replace("T", " ").split(".")
[0];
  const logMessage = `${timestamp} - ${level} - ${message}`;
  console.log(logMessage);
  try {
    fs.appendFileSync(LOG_FILE, logMessage + os.EOL, { encoding:
"utf-8" });
  } catch (e) {
    console.error("Ошибка при записи в лог:", e.message);
  }
}

// Чтение конфигурации
if (!fs.existsSync(CONFIG_FILE)) {
  log("Ошибка: конфигурационный файл не найден. Выход.", "ERROR");
  process.exit(1);
}

let config;
try {
  const raw = fs.readFileSync(CONFIG_FILE);
  config = JSON.parse(raw);
} catch (e) {
  log(`Ошибка при чтении конфигурации: ${e.message}. Выход.`, "ERROR");
  process.exit(1);
}

// Проверка параметров
const PORT = parseInt(config.port);
const HOST = config.ip_addres;
const INTERVAL = parseInt(config.time_to_wait) * 1000;

if (!PORT || !HOST || !INTERVAL) {
  log("Ошибка: неверные параметры конфигурации. Выход.", "ERROR");
  process.exit(1);
}

// Создание сокета
const client = new net.Socket();

client.connect(PORT, HOST, () => {
  log(`Успешное подключение к серверу: ${HOST}:${PORT}`);

  setInterval(() => {
    if (!client.destroyed) {

```

```

        try {
            client.write(MESSAGE);
            log(`Отправлено сообщение: '${MESSAGE}'`);
        } catch (e) {
            log(`Ошибка при отправке данных: ${e.message}`, "ERROR");
            client.destroy();
        }
    }
    else {
        log("Отключен от сервера.")
        client.destroy();
    }
}, INTERVAL);
});

client.on("data", (data) => {
    const text = data.toString().trim();
    console.log(text);
    log(`Получено сообщение: '${text}'`);
});

client.on("error", (err) => {
    log(`Ошибка соединения: ${err.message}`, "ERROR");
    process.exit(0); // Завершение программы
});

client.on("close", () => {
    log("Соединение с сервером закрыто", "INFO");
    process.exit(0); // Завершение программы
});

```

Конфигурационные файлы программ.

Синхронный сервер (config.json):

```
{  
  "port": 6969,  
  "ip_addres": "",  
  "handler_delay": 3,  
  "working_time": 12  
}
```

Асинхронный сервер (config.json):

```
{  
  "port": 6969,  
  "ip_addres": "",  
  "handler_delay": 3,  
  "working_time": 15  
}
```

Клиент (config.json):

```
{  
  "ip_addres": "192.168.3.69",  
  "port": 6969,  
  "time_to_wait": 5  
}
```

Асинхронный сервер (server.log):

12

2025-05-04 20:15:10 - INFO - Сервер остановлен вручную.

Клинет асинхронного сервера 0 (nyarch.log):

2025-05-04 17:14:42 - INFO - Успешное подключение к серверу: 192.168.3.69:6969

2025-05-04 17:14:47 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'

2025-05-04 17:14:50 - INFO - Получено сообщение: '42-ВБ521-О3М чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'

2025-05-04 17:14:52 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'

2025-05-04 17:14:55 - INFO - Получено сообщение: '42-ВБ521-О3М чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'

2025-05-04 17:14:57 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'

2025-05-04 17:15:00 - INFO - Получено сообщение: '42-ВБ521-О3М чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'

2025-05-04 17:15:00 - INFO - Соединение с сервером закрыто

Клинет асинхронного сервера 1 (windows.log):

2025-05-04 17:14:40 - INFO - Успешное подключение к серверу: 192.168.3.69:6969

2025-05-04 17:14:45 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'

2025-05-04 17:14:49 - INFO - Получено сообщение: '42-ВБ521-О3М чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'

2025-05-04 17:14:50 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'

2025-05-04 17:14:54 - INFO - Получено сообщение: '42-ВБ521-О3М чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'

2025-05-04 17:14:55 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'

2025-05-04 17:14:56 - INFO - Соединение с сервером закрыто

Клинет асинхронного сервера 2 (log_laptop.log):

2025-05-04 17:14:46 - INFO - Успешное подключение к серверу: 192.168.3.69:6969

2025-05-04 17:14:51 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'

2025-05-04 17:14:54 - INFO - Получено сообщение: '42-ВБ521-О3М чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'

2025-05-04 17:14:56 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'

2025-05-04 17:14:59 - INFO - Получено сообщение: '42-ВБ521-О3М чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'

2025-05-04 17:15:01 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'

2025-05-04 17:15:01 - INFO - Соединение с сервером закрыто

Клинет асинхронного сервера 3 (logs_sm-a12.log):

2025-05-04 17:14:45 - INFO - Успешное подключение к серверу: 192.168.3.69:6969

2025-05-04 17:14:50 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'

2025-05-04 17:14:53 - INFO - Получено сообщение: '42-ВБ521-О3М чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'

2025-05-04 17:14:55 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'

2025-05-04 17:14:58 - INFO - Получено сообщение: '42-ВБ521-О3М чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'

2025-05-04 17:15:00 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'

2025-05-04 17:15:00 - INFO - Соединение с сервером закрыто

Клинет асинхронного сервера 4 (logs_sm-a536x.log):

2025-05-04 17:14:46 - INFO - Успешное подключение к серверу: 192.168.3.69:6969

2025-05-04 17:14:51 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'

2025-05-04 17:14:54 - INFO - Получено сообщение: '42-B521-03M чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'
2025-05-04 17:14:56 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'
2025-05-04 17:14:59 - INFO - Получено сообщение: '42-B521-03M чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'
2025-05-04 17:15:01 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'
2025-05-04 17:15:01 - INFO - Соединение с сервером закрыто

Синхронный сервер (server.log):

2025-05-04 20:36:03 - INFO - Сервер запущен. Ожидание сообщений...
2025-05-04 20:36:26 - INFO - Установлено соединение с клиентом: ('127.0.0.1', 45892)
2025-05-04 20:36:31 - INFO - Получено сообщение от клиента ('127.0.0.1', 45892): 'Егоров Александр Владиславович М3О-125БВ-24'
2025-05-04 20:36:34 - INFO - Сообщение отправлено клиенту ('127.0.0.1', 45892): '42-B521-03M чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'
2025-05-04 20:36:36 - INFO - Получено сообщение от клиента ('127.0.0.1', 45892): 'Егоров Александр Владиславович М3О-125БВ-24'
2025-05-04 20:36:39 - INFO - Сообщение отправлено клиенту ('127.0.0.1', 45892): '42-B521-03M чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'
2025-05-04 20:36:39 - INFO - Клиент отключен от сервера: ('127.0.0.1', 45892)
2025-05-04 20:36:39 - INFO - Установлено соединение с клиентом: ('192.168.3.47', 36004)
2025-05-04 20:36:39 - INFO - Получено сообщение от клиента ('192.168.3.47', 36004): 'Егоров Александр Владиславович М3О-125БВ-24Егоров Александр Владиславович М3О-125БВ-24'
2025-05-04 20:36:42 - INFO - Сообщение отправлено клиенту ('192.168.3.47', 36004): '42-B521-03M чивовалсидалВ рднаскелА ворогЕ42-B521-03M чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'
2025-05-04 20:36:42 - INFO - Получено сообщение от клиента ('192.168.3.47', 36004): 'Егоров Александр Владиславович М3О-125БВ-24'
2025-05-04 20:36:45 - INFO - Сообщение отправлено клиенту ('192.168.3.47', 36004): '42-B521-03M чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'
2025-05-04 20:36:46 - INFO - Получено сообщение от клиента ('192.168.3.47', 36004): 'Егоров Александр Владиславович М3О-125БВ-24'
2025-05-04 20:36:49 - INFO - Сообщение отправлено клиенту ('192.168.3.47', 36004): '42-B521-03M чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'
2025-05-04 20:36:51 - INFO - Клиент отключен от сервера: ('192.168.3.47', 36004)

Клинет синхронного сервера 0 (logs.log):

2025-05-04 17:36:26 - INFO - Успешное подключение к серверу: 192.168.3.69:6969
2025-05-04 17:36:31 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'
2025-05-04 17:36:34 - INFO - Получено сообщение: '42-B521-03M чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'
2025-05-04 17:36:36 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'
2025-05-04 17:36:39 - INFO - Получено сообщение: '42-B521-03M чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'
2025-05-04 17:36:39 - INFO - Соединение с сервером закрыто

Клинет синхронного сервера 1 (log_laptop.log):

2025-05-04 17:36:26 - INFO - Успешное подключение к серверу: 192.168.3.69:6969
2025-05-04 17:36:31 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'
2025-05-04 17:36:36 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'
2025-05-04 17:36:41 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'
2025-05-04 17:36:42 - INFO - Получено сообщение: '42-B521-03M чивовалсидалВ рднаскелА ворогЕ42-B521-03M чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'
2025-05-04 17:36:45 - INFO - Получено сообщение: '42-B521-03M чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'
2025-05-04 17:36:46 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович М3О-125БВ-24'
2025-05-04 17:36:49 - INFO - Получено сообщение: '42-B521-03M чивовалсидалВ рднаскелА ворогЕ. Сервер написан Егоровым А.В. М3О-125БВ-24'

2025-05-04 17:36:51 - INFO - Отправлено сообщение: 'Егоров Александр Владиславович МЗО-125БВ-24'
2025-05-04 17:36:51 - INFO - Соединение с сервером закрыто

Описание работы программы

Этот проект реализует клиент-серверное приложение для обмена сообщениями через TCP/IP с использованием сокетов. Сервер написан на Python, клиент — на Node.js. Оба компонента ведут логи для записи ключевых событий. Сервер поддерживает как синхронную, так и асинхронную обработку запросов с использованием разных архитектур, что позволяет эффективно обрабатывать несколько клиентов одновременно.

Общая архитектура:

1. Сервер (Python):

- Запускается и ожидает входящие подключения от клиентов на заданном порту. При каждом подключении создается обработчик (асинхронный или синхронный) для работы с клиентом.
- Для каждого клиента выполняется процесс получения сообщения, его обработки (с задержкой) и отправки ответа.
- Для асинхронной обработки используется `asyncio`, а для синхронной — стандартная библиотека сокетов.
- Ведет подробное логирование событий и взаимодействий с клиентами.
- После заданного времени работы сервер автоматически отключает клиентов.

2. Клиент (node.js):

- Подключается к серверу и периодически отправляет информацию о студенте (ФИО и группа). Ожидает ответ от сервера.
- Ведет логирование событий подключения, отправки и получения сообщений.

Подробное описание компонентов:

1. Сервер (python):

- **Импорты:** Используются стандартные модули:
 - `socket`: Для работы с сетевыми сокетами.
 - `time`: Для задержки в работе сервера.
 - `json`: Для чтения конфигурационных параметров.
 - `logging`: Для логирования событий.
 - `asyncio`: Для асинхронной обработки запросов.

- **Конфигурация:** Сервер считывает параметры из конфигурационного файла `config.json`, который включает:
 - `port`: Номер порта для прослушивания.
 - `ip_address`: IP-адрес сервера.
 - `handler_delay`: Задержка для имитации обработки.
 - `working_time`: Время работы сервера.
- **Обработка клиентов:** Для каждого клиента запускается отдельная асинхронная задача или поток:
 - Получает сообщение от клиента.
 - Обработывает его с задержкой.
 - Отправляет обратно измененное сообщение.
 - Логирует процесс получения и отправки сообщений.
- **Запуск сервера:**
 - Создается сокет для прослушивания входящих подключений.
 - Используется асинхронный цикл для обработки клиентов.

2. Клиент (node.js):

- **Импорты:** Используются модули Node.js:
 - `net`: Для создания TCP-сокета.
 - `fs`: Для работы с файловой системой (чтение конфигурации и запись логов).
 - `os`: Для работы с операционной системой.
- **Конфигурация:** Клиент считывает параметры из файла `config.json`, включая:
 - `ip_address`: IP-адрес сервера.
 - `port`: Порт сервера.
 - `time_to_wait`: Интервал между отправками сообщений.
 - `working_time`: Время работы клиента.
- **Логирование:** Клиент ведет журнал всех операций, используя временные метки для логов.
- **Процесс работы:**
 - Подключается к серверу и периодически отправляет строку с информацией о студенте.

- Ожидает ответ и логирует его.
- Клиент завершает свою работу по истечении заданного сервером времени.

Взаимодействие:

- Клиент устанавливает TCP-соединение с сервером, отправляет данные, получает ответ и продолжает цикл обмена информацией.
- Сервер обрабатывает данные клиента, имитируя работу, и отправляет обратно измененное сообщение.
- Для каждого клиента сервер использует либо асинхронные задачи (через `asyncio`), либо обычные синхронные потоки (в зависимости от запущенного файла).
- Клиент завершает свою работу по истечении заданного времени или при возникновении ошибок.

Вывод по работе:

В ходе выполнения работы был успешно реализован и протестирован клиент-серверный обмен сообщениями по протоколу TCP/IP с использованием сокетов. Разработаны две версии серверной части: асинхронная и синхронная. Клиентская часть взаимодействует с сервером на Node.js, отправляя данные, получая ответы и корректно завершает свою работу по завершении заданного времени.

Ссылки:

Все файлы можно посмотреть на github:

https://github.com/swrneko/mai_shit/tree/main/1lvl_2sem/web/laba_2