МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет прикладной математики и информатики

Кафедра ТПИ

Дисциплина: «Сетевые информационные технологии»

Лабораторная работа №4

ПРОТОКОЛ ПЕРЕДАЧИ ФАЙЛОВ FTP

Факультет: ФПМИ

Группа: ПМИМ-31

Студенты: Тарулин М.А., Холодова В.С.

Преподаватель: Кобылянский В.Г.

Дата выполнения:

Отметка о защите:

Новосибирск, 2024 г.

1. Цель работы

Целью работы является изучение принципов организации взаимодействия прикладных программ с помощью протокола передачи данных FTP и приобретение практических навыков создания клиентских приложений, использующих протокол FTP.

2. Задание

2.1. Создать на локальном компьютере каталог Lab3.

2.2. Запустить анализатор Wireshark для перехвата FTP-трафика.

2.3. С помощью браузера подключится по протоколу FTP к серверам fpm2.ami.nstu.ru и ftp.nstu.ru. Скачать из домашнего каталога на сервере fpm2.ami.nstu.ru файл, имя которого задано в таблице 2.5 лабораторной работы №2. По захваченным данным построить диаграмму потоков.

2.4. Запустить стандартный ftp-клиент Windows и выполнить действия:

- Вывести список поддерживаемых команд;
- Подключится к серверу fpm2.ami.nstu.ru;
- Создать на локальном компьютере в каталоге Lab3 подкаталоги Lab3_1 и Lab3_2;
- Скачать в подкаталог Lab3_1 несколько файлов из домашнего каталога на сервере;
- Просмотреть содержимое каталога Lab3_1 путем временного перехода в командную строку Windows;
- Изучить способы навигации по файловой системе сервера;
- Создать в домашнем каталоге сервера подкаталог FTP и скопировать в него с локального компьютера файл, содержащий отчет по лабораторной работе №2;
- По захваченным данным построить диаграмму потоков, сравнить ее с диаграммой из п.3.

2.5. Разработать программу, реализующие следующие функции ftp-клиента:

- Создание и удаление каталогов;
- Перемещение по каталогам;
- Вывод содержимого каталога;
- Получение, отправка, удаление и переименование файлов;

2.6. С помощью разработанной программы выполнить указанные действия, используя локальный каталог Lab3_2:

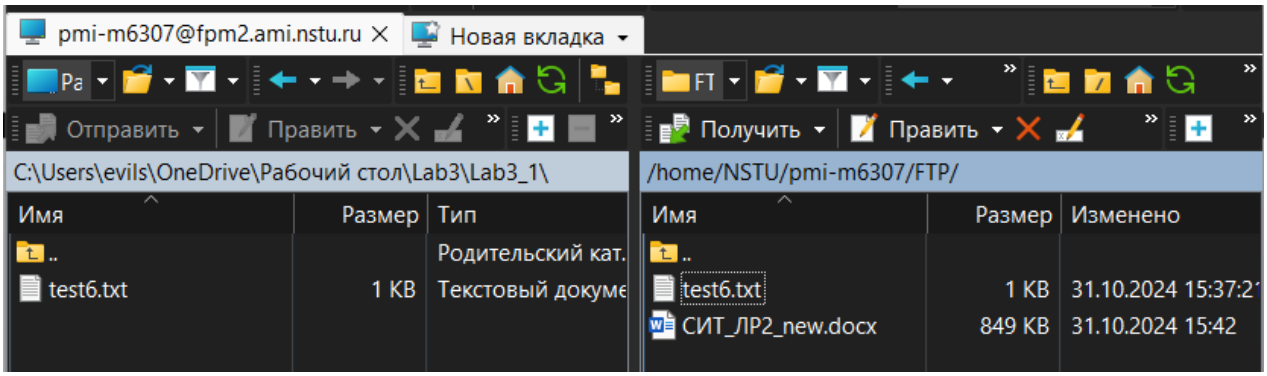
- Провести сеанс обмена файлами с любым доступным ftp-сервером;
- По захваченным данным построить диаграмму потоков, сравнить ее с ранее построенными диаграммами.

3. Ход работы

3.1. Создадим на локальном компьютере каталог Lab3.

3.2. Запустим анализатор трафика Wireshark для перехвата FTP-трафика.

3.3. С помощью браузера подключимся по протоколу FTP к серверам fpm2.ami.nstu.ru и ftp.nstu.ru. Скачаем из домашнего каталога на сервере fpm2.ami.nstu.ru файл test6.txt. На рис. 1 изобразим результат выполнения задания, на рис. 2 изобразим диаграмму потоков для перехваченного сеанса.



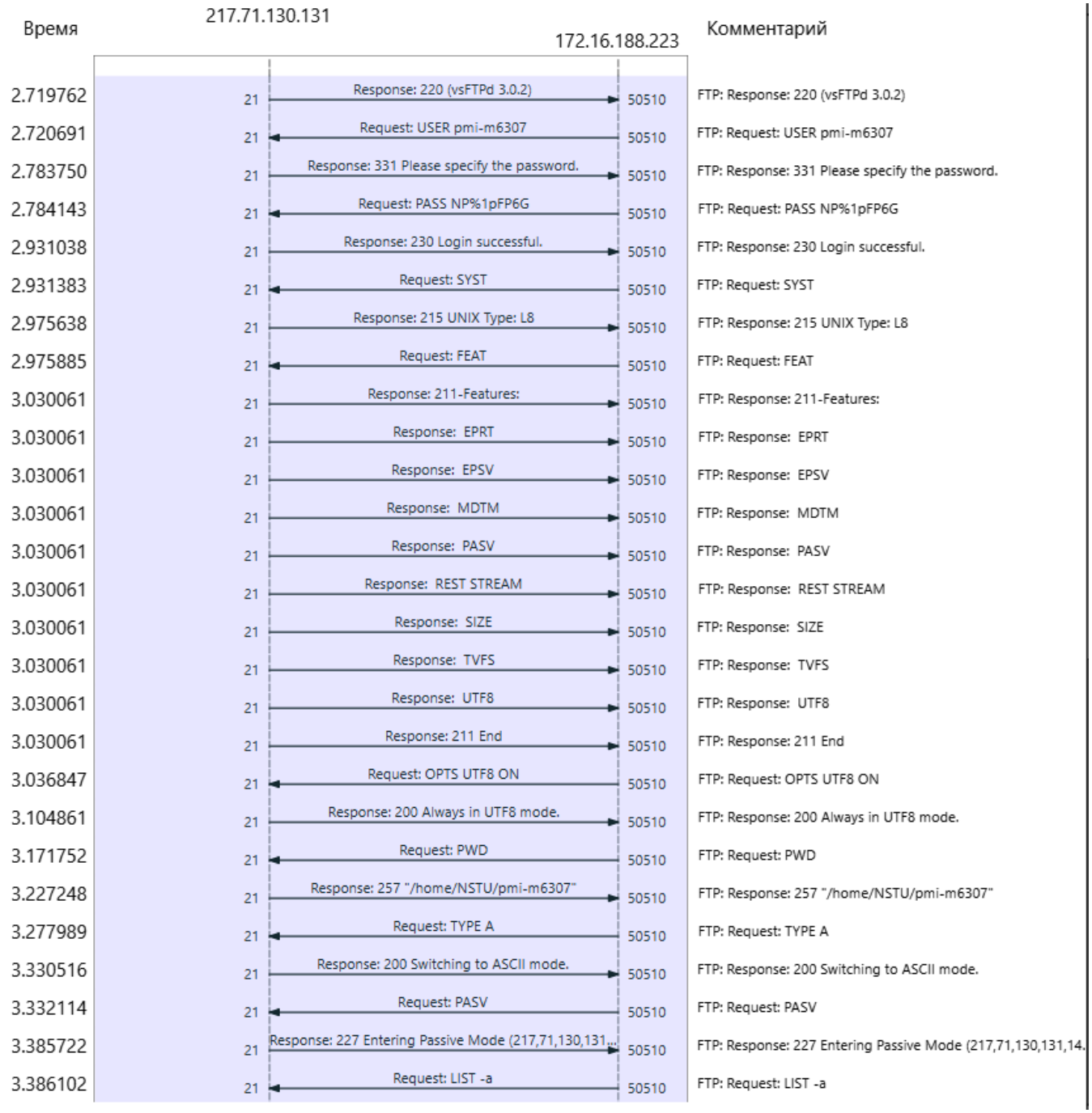


Рисунок 2 – Диаграмма потоков.

Рисунок 2 – Диаграмма потоков (продолжение).

3.4. Запустим стандартный ftp-клиент Windows и выполним действия:

- Выведем список поддерживаемых команд;
- Подключимся к серверу fpm2.ami.nstu.ru;
- Создадим на локальном компьютере в каталоге Lab3 подкаталоги Lab3_1 и Lab3_2;
- Скачаем в подкаталог Lab3_1 несколько файлов из домашнего каталога на сервере;
- Просмотрим содержимое каталога Lab3_1 путем временного перехода в командную строку Windows;
- Создадим в домашнем каталоге сервера подкаталог FTP и скопируем в него с локального компьютера файл, содержащий отчет по лабораторной работе №2;
- Построим по захваченным данным диаграмму потоков и сравним с рис. 1.

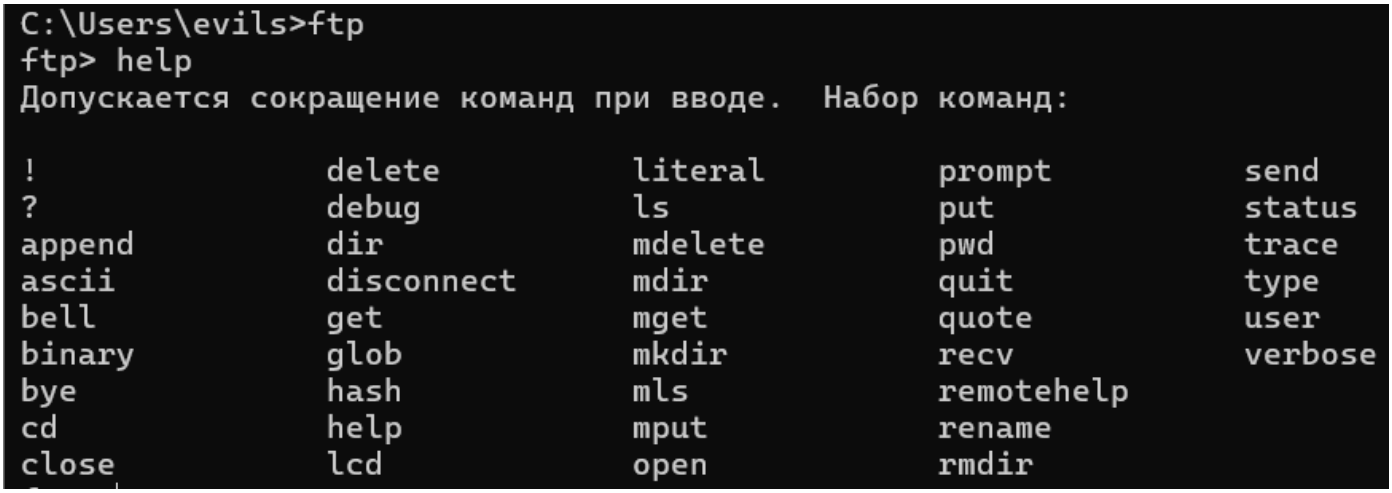


Рисунок 3.1 – Вывод списка поддерживаемых команд для стандартного ftp-клиента Windows.

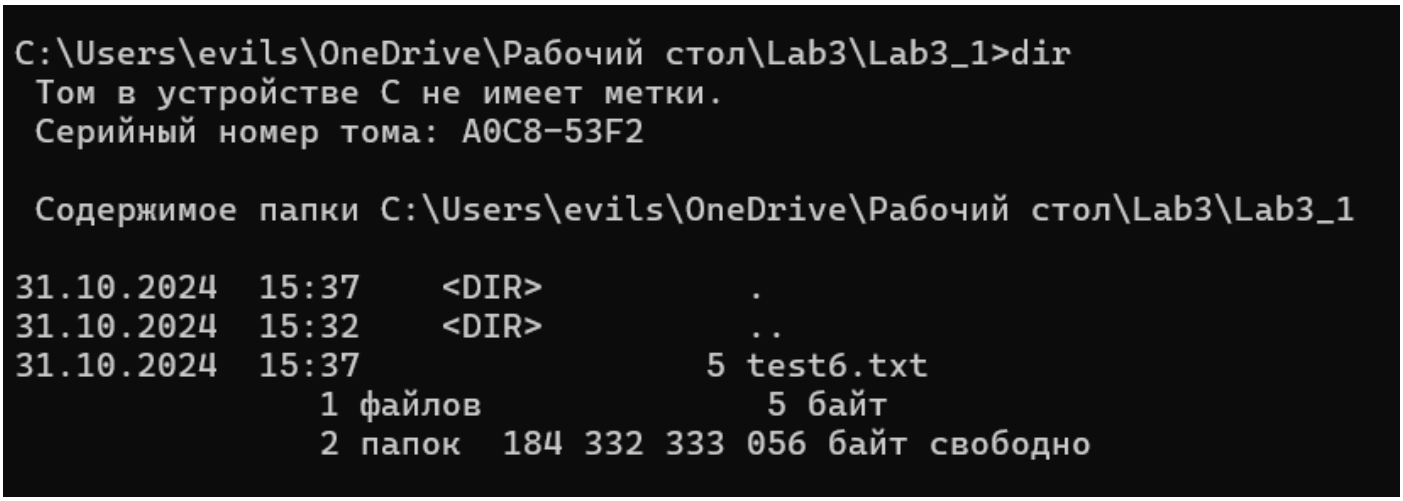


Рисунок 3.2 – Вывод содержимого каталога Lab3_1 после загрузки файла test6.txt.

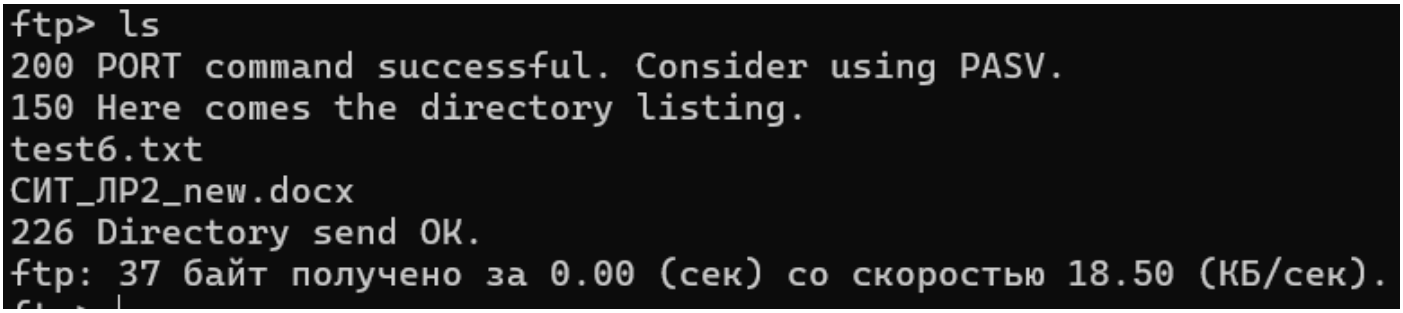


Рисунок 3.3 – Вывод содержимого каталога сервера после загрузки файла с отчетом по ЛР2.

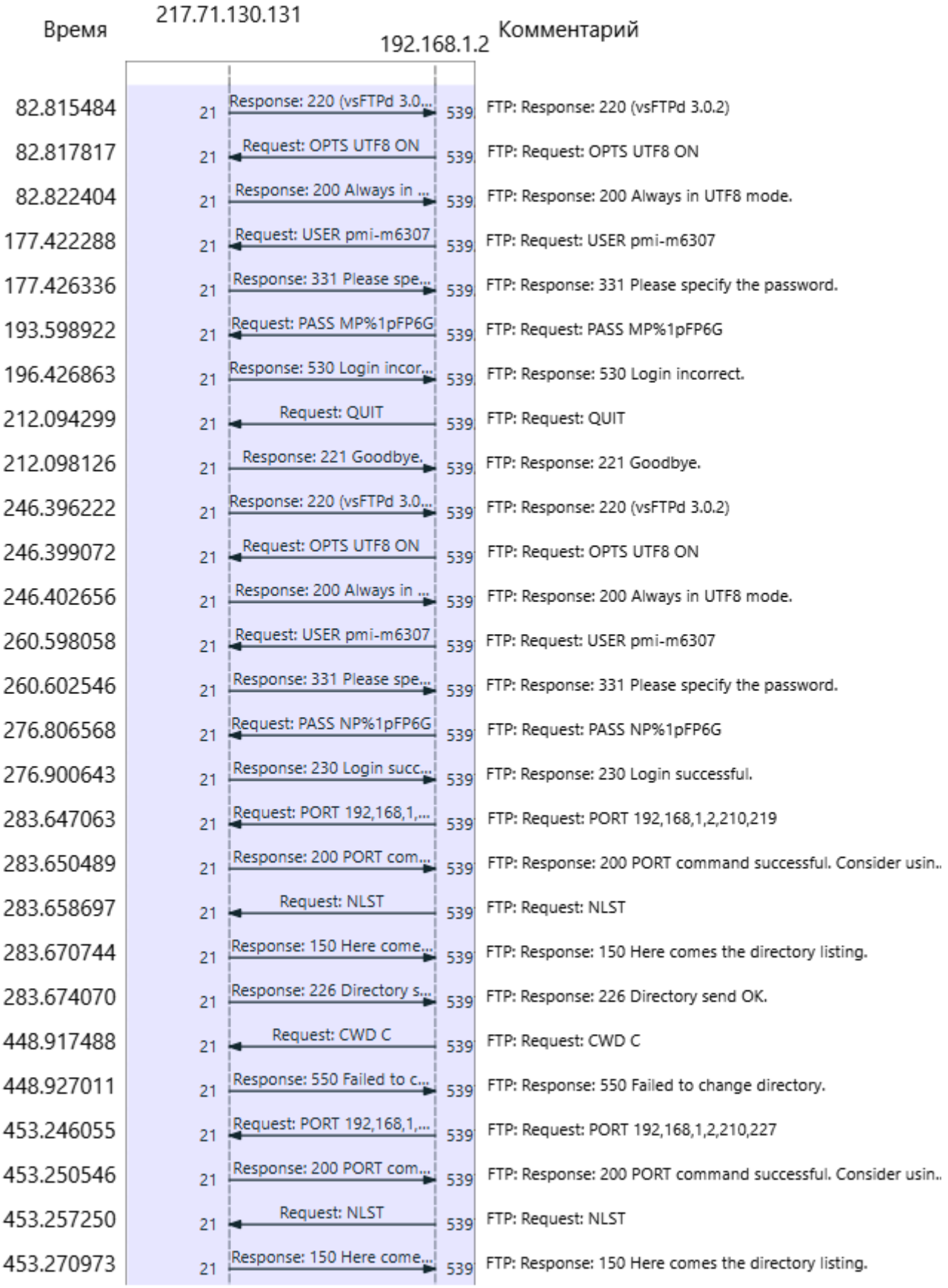


Рисунок 4 – Диаграмма потоков при использовании стандартного ftp-клиента Windows.

Рисунок 4 – Диаграмма потоков при использовании стандартного ftp-клиента Windows (продолжение).

В первом сеансе пользователь только ввел имя пользователя pmi-m6307 и пароль PASS NP%1pFP6G, после чего программа WinSCP автоматически выполнила оставшиеся команды. Она настроила передачу данных в режиме UTF-8 через команду OPTS UTF8 ON, перешла в текстовый режим передачи файлов с помощью TYPE A, получила информацию о типе операционной системы сервера с помощью команды

SYST и список поддерживаемых функций через FEAT. Кроме того, протокол переключился в пассивный режим. Далее через интерфейс приложения был загружен файл test6.txt. Программа выполнила команды CWD, проверила размер и дату последнего изменения этого файла с помощью команд SIZE и MDTM, а также загрузила файл test6.txt с использованием команды RETR.

Во втором сеансе используется стандартная утилита командной строки ftp в Windows. Первая попытка входа с неверным паролем приводит к ошибке, но после повторного успешного входа создается новый каталог FTP с командой XMKD FTP, производится переход в этот каталог с помощью CDW FTP и загружается туда файл СИТ_ЛР2.docx с использованием STOR. В отличие от первого сеанса, здесь не проводится проверка размеров и дат изменения файлов, а также переходов в другие каталоги кроме созданного.

3.5. Разработаем программу, реализующие создание и удаление, перемещение по каталогам, вывод содержимого каталога, получение, отправка, удаление и переименование файлов ftp-клиента. Результат работы программы продемонстрируем на рис. 5.1-5.8.

```
C:\Users\evils\PycharmProjects\FTP_client\.venv\Scripts\python.exe C:\Users\evils\PycharmProjects\FTP_client\main.py
Введите ftp адресс: fpm2.ami.nstu.ru
Введите логин: pmi-m6307
Введите пароль: NP%1pFP6G
Соединение установлено
Список доступных комманд:
1. Создать каталог на сервере
2. Удалить каталог на сервере
3. Вывести содержимое каталога
4. Сменить каталог
5. Получить файл
6. Загрузить файл
7. Переименовать файл
8. Удалить файл
0. Выход из программы
Выберите команду: 1
Введите имя создаваемого каталога: test_dir
Каталог test_dir успешно создан
```

Рисунок 5.1 – Функционал создания каталога.

```
Выберите команду: 2
Введите имя удаляемого каталога: test_dir
Каталог test_dir успешно удален
```

Рисунок 5.2 – Функционал удаления каталога.

```
Выберите команду: 3
drwxr-xr-x    2 33345    3031         4096 Oct 31 08:42 FTP
drwxr-xr-x    2 33345    3031         4096 Nov 01 13:52 FTP_2
drwxr-xr-x    2 33345    3031         4096 Nov 03 11:46 FTP_3
drwxr-xr-x    2 33345    3031         4096 Nov 03 11:51 FTP_4
drwxr-xr-x    2 33345    3031         4096 Nov 03 17:17 FTP_5
-rw-r--r--    1 33345    3031            5 Oct 28 09:10 test9.txt
None
```

Рисунок 5.3 – Функционал просмотра содержимого каталога.

```
Выберите команду: 4
Введите имя каталога, в который хотите переместиться: FTP
Выберите команду: 3
-rw-r--r--    1 33345    3031       868960 Oct 31 08:42 СИТ_ЛР2.docx
None
```

Рисунок 5.4 – Функционал перемещения по каталогам.

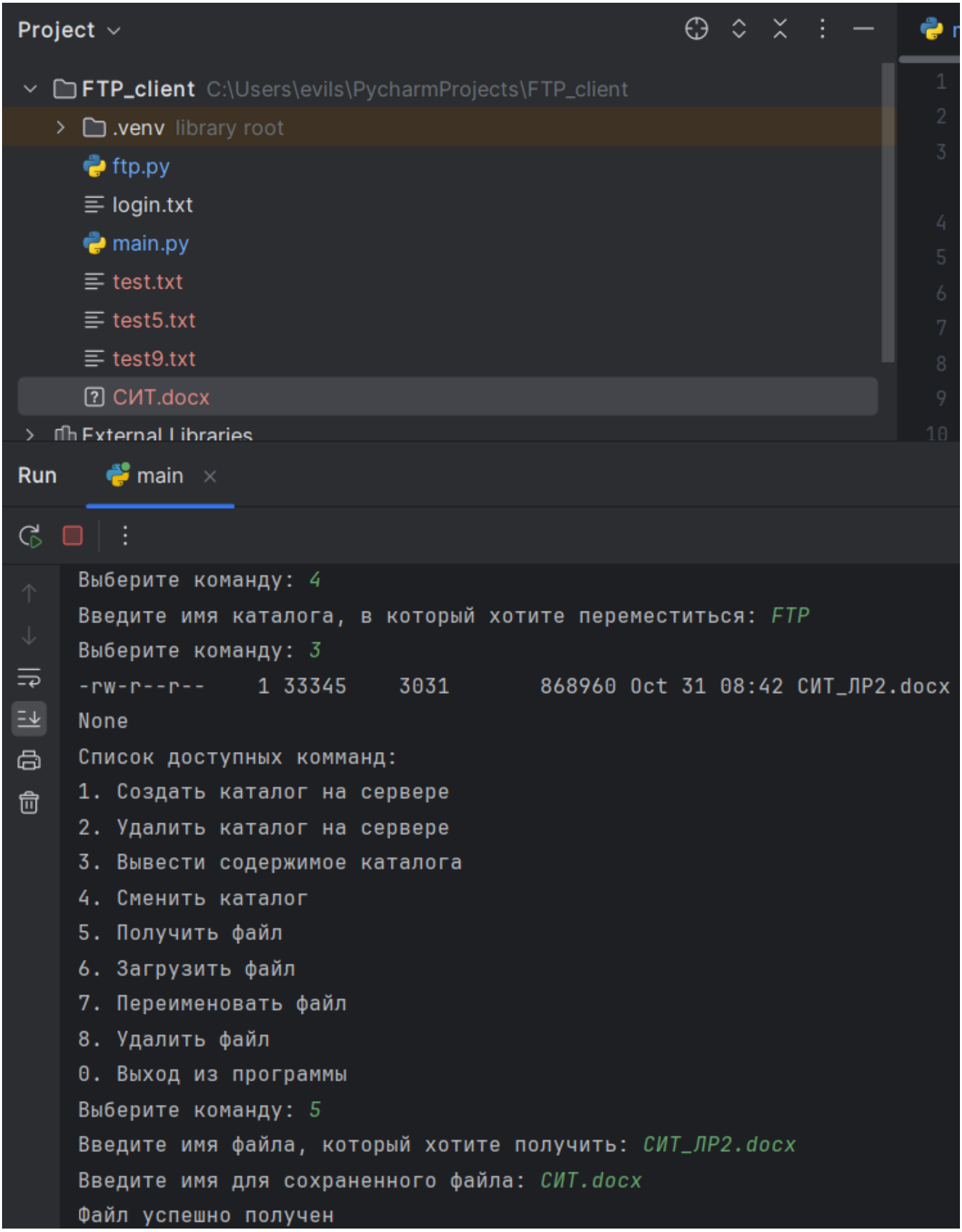


Рисунок 5.5 – Функционал загрузки файлов с сервера.

```
Выберите команду: 6
Введите имя файла, который хотите загрузить: СИТ.docx
Введите имя для загружаемого файла: СИТ.docx
Файл успешно загружен
Выберите команду: 3
-rw-r--r--    1 33345    3031       868960 Nov 04 09:44 СИТ.docx
-rw-r--r--    1 33345    3031       868960 Oct 31 08:42 СИТ_ЛР2.docx
None
```

Рисунок 5.6 – Функционал загрузки файлов на сервер.

```
Выберите команду: 7
Введите имя файла, который хотите переименовать: СИТ_ЛР2.docx
Введите новое имя для файла: СИТ_ЛР2_new.docx
Файл успешно переименован с "СИТ_ЛР2.docx" на "СИТ_ЛР2_new.docx"
Выберите команду: 3
-rw-r--r--    1 33345    3031       868960 Nov 04 09:44 СИТ.docx
-rw-r--r--    1 33345    3031       868960 Oct 31 08:42 СИТ_ЛР2_new.docx
None
```

Рисунок 5.7 – Функционал переименования файла на сервере.

```
Выберите команду: 8
Введите имя файла, который хотите удалить: СИТ.docx
Файл СИТ.docx успешно удален
Выберите команду: 3
-rw-r--r--    1 33345    3031       868960 Oct 31 08:42 СИТ_ЛР2_new.docx
None
```

Рисунок 5.8 – Функционал удаления файла на сервере.

По результатам выполнения задания с помощью разработанной программы построим диаграмму потоков.

Рисунок 6 – Диаграмма потоков при использовании разработанной программы.

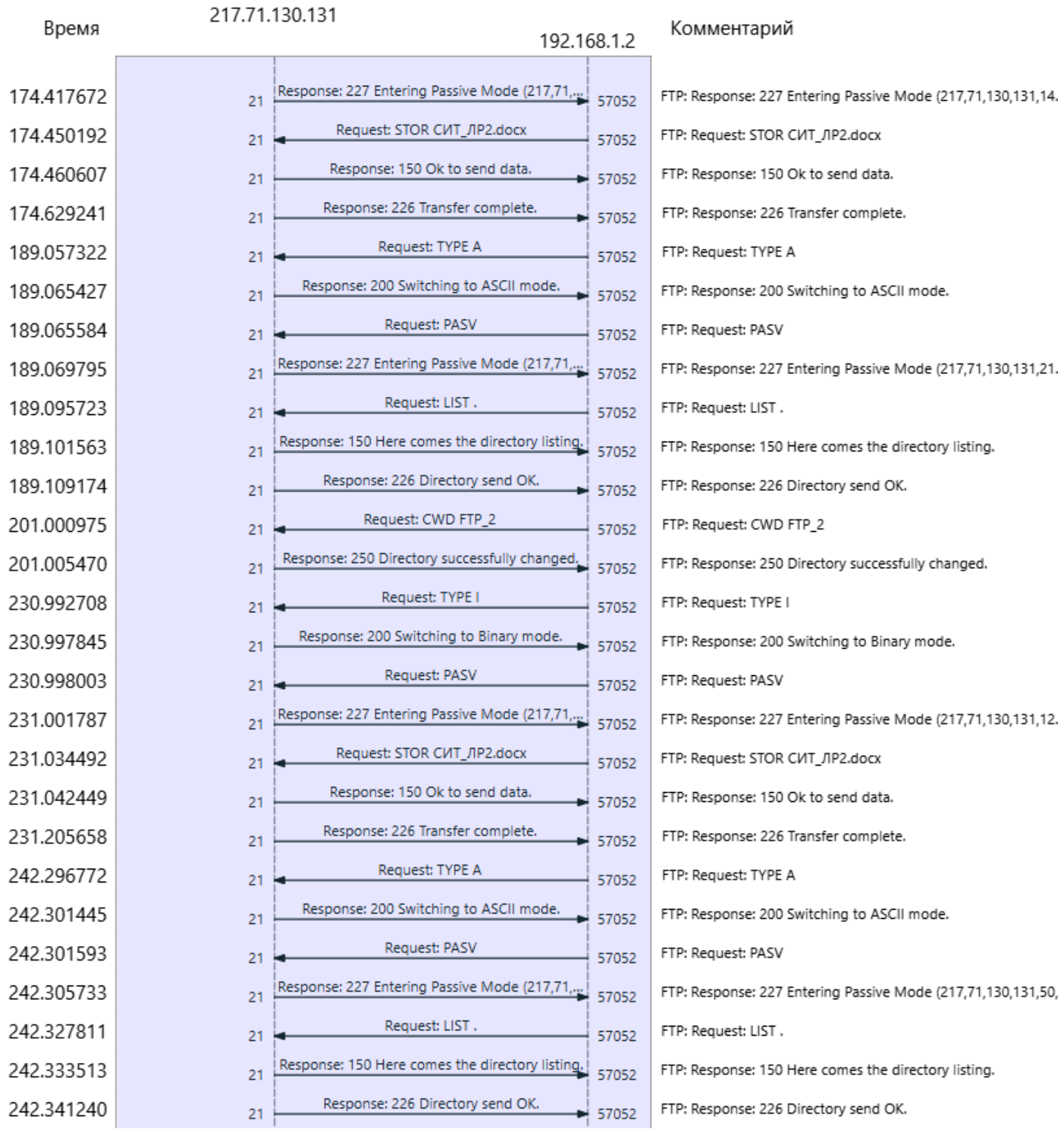


Рисунок 6 - Диаграмма потоков при использовании разработанной программы (продолжение).

Диаграмма потоков при работе с FTP-сервером с помощью разработанной программы отличается от диаграммы потока, полученной при использовании стандартного ftp-клиента Windows. При использовании разработанного приложения происходит переход в пассивный режим перед каждой операцией передачи данных, аналогично, как при использовании приложения WinSCP.

4. Вывод

В ходе выполнения лабораторной работы была изучена организация взаимодействия прикладных программ с использованием протокола передачи данных FTP. Были приобретены практические навыки создания клиентских приложений, использующих протокол FTP. В рамках работы были выполнены следующие задачи: проведен анализ FTP-трафика с помощью Wireshark, разработано клиентское приложение для работы с FTP-сервером, реализовано взаимодействие с различными FTP-серверами. Было проведено сравнение диаграмм потоков, полученных при использовании различных FTP-клиентов. В результате работы было установлено, что несмотря на различия в интерфейсах и функциональности различных FTP-клиентов, основные принципы взаимодействия с FTP-сервером остаются неизменными.

5. Код программы

main.py

```
from ftp import FtpServer


def command_list():
    print('Список доступных комманд:')
    print('1. Создать каталог на сервере')
    print('2. Удалить каталог на сервере')
    print('3. Вывести содержимое каталога')
    print('4. Сменить каталог')
    print('5. Получить файл')
    print('6. Загрузить файл ')
    print('7. Переименовать файл')
    print('8. Удалить файл')
    print('0. Выход из программы')


def main_menu(ftp_connection: FtpServer) -> None:
    command_list()

    while True:
        command = input('Выберите команду: ')
        if command == '1':
            dir_name = input('Введите имя создаваемого каталога: ')
            ftp_connection.make_directory(dir_name)
            command_list()

        elif command == '2':
            dir_name = input('Введите имя удаляемого каталога: ')
            ftp_connection.delete_directory(dir_name)
            command_list()

        elif command == '3':
            ftp_connection.show_directory_contents()
```

```
            command_list()

        elif command == '4':
            dir_name = input('Введите имя каталога, в который хотите переместиться: ')
            ftp_connection.change_directory(dir_name)

        elif command == '5':
            filename = input('Введите имя файла, который хотите получить: ')
            local_filename = input('Введите имя для сохраненного файла: ')
            ftp_connection.download_file(filename, local_filename)

        elif command == '6':
            filename = input('Введите имя файла, который хотите загрузить: ')
            local_filename = input('Введите имя для загружаемого файла: ')
            ftp_connection.upload_file(filename, local_filename)

        elif command == '7':
            filename = input('Введите имя файла, который хотите переименовать: ')
            new_filename = input('Введите новое имя для файла: ')
            ftp_connection.rename_file(filename, new_filename)

        elif command == '8':
            filename = input('Введите имя файла, который хотите удалить: ')
            ftp_connection.delete_file(filename)

        elif command == '0':
            return None

        else:
            print('Введенная команда не существует')
            command_list()


if __name__ == '__main__':
    SERVER_ADDRESS = input("Введите ftp адресс: ")
    USERNAME = input("Введите логин: ")
    PASSWORD = input("Введите пароль: ")

    try:
        ftp = FtpServer(SERVER_ADDRESS, USERNAME, PASSWORD)
        main_menu(ftp)

        ftp.close()
    except Exception as e:
        print(f'Ошибка: {e}')
```

ftp.py

```
from ftplib import FTP
import socket


class FtpServer:
    def __init__(self, ftp_address: str, username: str, password: str):
        self.ftp_address = ftp_address
        self.username = username
        self.password = password
```

```
    self.connection = self.connect()

def connect(self) -> FTP | None:
    try:
        ftp = FTP(timeout=10)
        ftp.connect(self.ftp_address)

        ftp.login(user=self.username, passwd=self.password)
        print('Соединение установлено')
        return ftp

    except socket.gaierror:
        print(f'Ошибка: Невозможно разрешить адрес {self.ftp_address}')

    except socket.error as e:
        print(f'Ошибка сокета: {e}')

    except ConnectionRefusedError:
        print(f'Подключение к {self.ftp_address} отклонено')

    except TimeoutError:
        print(f'Превышено время ожидания при подключении к {self.ftp_address}')

    except Exception as e:
        print(f'Ошибка при подключении: {e}')

    return None

def make_directory(self, directory_name: str) -> None:
    self.connection.mkd(directory_name)
    print(f'Каталог {directory_name} успешно создан')

def delete_directory(self, directory_name: str) -> None:
    self.connection.rmd(directory_name)
    print(f'Каталог {directory_name} успешно удален')

def show_directory_contents(self) -> None:
    print(self.connection.dir('.'))

def change_directory(self, directory_name: str) -> None:
    self.connection.cwd(directory_name)

def download_file(self, filename: str, local_filename: str) -> None:
    with open(local_filename, 'wb') as local_file:
        self.connection.retrbinary(f'RETR {filename}', local_file.write)
    print('Файл успешно получен')

def upload_file(self, filename: str, local_filename: str) -> None:
    with open(local_filename, 'rb') as local_file:
        self.connection.storbinary(f'STOR {filename}', local_file)
    print('Файл успешно загружен')

def rename_file(self, filename: str, new_filename: str) -> None:
    self.connection.sendcmd(f'RNFR {filename}')
    self.connection.sendcmd(f'RNTO {new_filename}')
```

```
    print(f'Файл успешно переименован с "{filename}" на "{new_filename}"')

def delete_file(self, filename: str) -> None:
    self.connection.delete(filename)
    print(f'Файл {filename} успешно удален')

def close(self):
    self.connection.close()
    print('Соединение разорвано')
```