Учреждение образования

«Белорусский Государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

«Многопоточный FTP-Клиент»

по дисциплине

«Системное программное обеспечение вычислительных машин»

Выполнил: Руководитель:

студент гр. 850505

Григарович Алексей Андреевич Глоба А. А.

Минск, 2020

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc41505934)

[1. ОБЗОР ИСТОЧНИКОВ 4](#_Toc41505935)

[2. СТРУКТУРНОЕ ПРОЕКТИРОВАНИE 5](#_Toc41505936)

[3. ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ 6](#_Toc41505937)

[4. РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ 10](#_Toc41505938)

[5. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 11](#_Toc41505939)

[6. ТЕСТИРОВАНИЕ 13](#_Toc41505940)

[ЗАКЛЮЧЕНИЕ 14](#_Toc41505941)

[СПИСОК ИСТОЧНИКОВ 15](#_Toc41505942)

# **ВВЕДЕНИЕ**

В данной работе планируется реализация FTP-Клиента для упрощенного доступа к FTP-Серверу. В зависимости от назначения может либо предоставлять пользователю простой доступ к удалённому FTP-серверу в режиме текстовой консоли, беря на себя только работу по пересылке команд пользователя и файлов, либо отображать файлы на удалённом сервере, как если бы они являлись частью файловой системы компьютера пользователя, либо и то и другое. В этой работе будет использована текстовая консоль, так как в пользовательском интерфейсе нет необходимости.

В простейшем для пользователя случае FTP-клиент представляет собой эмулятор файловой системы, которая просто находится на другом компьютере. С этой файловой системой можно совершать все привычные пользователю действия: копировать файлы с сервера и на сервер, удалять файлы, создавать новые файлы. В отдельных случаях возможно также открытие файлов — для просмотра, запуска программ, редактирования. Необходимо учитывать лишь, что открытие файла подразумевает его предварительное скачивание на компьютер пользователя.

В работе будет использован чужой FTP-Сервер для тестирования собственного FTP-Клиента, так как цель данной работы написать FTP-Клиент, а не сервер.

# **ОБЗОР ИСТОЧНИКОВ**

Для разработки была выбрана IDE CLion, так как это по моему мнению лучшее IDE для дистрибутива Manjaro Linux и не только. Разработка будет идти на языке C++.

FileZilla — один из бесплатных FTP клиентов. Он поддерживает FTP, SFTP, и FTPS (FTP через SSL/TLS) и имеет настраиваемый интерфейс с поддержкой смены тем оформления. Оснащён возможностью перетаскивания объектов, синхронизацией каталогов и поиском на удалённом сервере. Поддерживает многопоточную загрузку файлов, а также докачку при обрыве (если поддерживается сервером) интернет-соединения. В этой же работе поддерживается многопоточность, а интерфейс и вовсе отсутствует.

<https://ru.wikipedia.org/wiki/FTP-клиент>

<https://ru.wikipedia.org/wiki/FTP>

[https://ru.wikipedia.org/wiki/Список\_кодов\_ответов\_FTP](https://ru.wikipedia.org/wiki/Список_кодов_ответов_FTP%20)

<https://support.solarwinds.com>

# **СТРУКТУРНОЕ ПРОЕКТИРОВАНИE**

* 1. **Общее структурное описание состава программного обеспечения**

Данная программа состоит из четырех модулей: модуль пользовательского ввода, управляющий модуль, модуль отправки команд, модуль передачи/принятия файлов.

Модуль пользовательского ввода отвечает за ввод информации непосредственно пользователем.

Управляющий модуль осуществляет обработку ввода и последующее управление оставшимися модулями.

Модуль отправки команд осуществляет отправку команд на сервер и получение ответа с сервера.

Модуль передачи/принятия файлов отвечает за загрузку и скачивание файлов с сервера.

* 1. **Описание деления проекта на отдельные программы, модули или другие составные части**

Модуль пользовательского ввода ждет пока пользователь введет команду в консоль, затем он ее отправляет в управляющий модуль

Управляющий модуль получает введенную пользователем команду от соответствующего модуля и анализирует ее. В зависимости от того какая команда была введена, он посылает команду модулю отправки команд. Если введенная команда не реализована, то он снова ожидает ввод.

Модуль отправки команд отправляет команду по сокету команд серверу, после чего принимает ответ от сервера и выводит его в консоль.

Модуль передачи/принятия файлов использует второй поток, в котором по сокету данных отправляет или получает файлы и записывает их на диск. Еще этот модуль используется для отображения файлов и папок на сервере, так как сон отправляет их имена по очереди по сокету данных.

# **ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ**

Диаграмма классов представлена в Приложении 1.

# **РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ**

Блок-схема представлена в Приложении 3.

Для примера рассмотрим алгоритм действий, при вводе пользователем команды “STOR”. Сперва отправляется команда “STOR аргумент” на сервер(аргумент – имя файла, которое мы хотим загрузить). После чего мы ждем ответ от сервера: если пришел ответ с кодом 125, значит что канал открыт, мы сразу просим пользователя ввести директорию файла, то есть где нам взять файл, название которого в аргументе, и открываем его одним из способов: бинарный или ascii. После открытия файла мы начинаем его данные последовательно записывать в буфер и отправлять по сокету данных на сервер, где он и обрабатывает получаемые данные и записывает в файл.

# **РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

При запуске программы пользователь увидит консоль. Первым делом программа попросит нас ввести ip адрес сервера, а затем порт для подключения к нему. После подключения к нему мы уже можем вводить команды, но не все. Сначала нужно “залогиниться” на сервере с помощью команд USER и PASS:

|  |  |
| --- | --- |
| USER имя\_пользователя | После USER нам нужно ввести свой логин на сервере |
| PASS пароль | Пароль нам нужен чтобы войти под логином и уже дальше производить нужные нам операции. |

После логина мы уже можем делать все доступные нам операции. Есть много команд для отправки на FTP-Сервер, но не все могут быть реализованы. В нашем же случае, на FTP-Сервере реализованы следующие команды:

|  |  |
| --- | --- |
| USER “аргумент” | Отправка логина серверу. Аргумент команды — логин |
| PASS “аргумент” | Отправка пароля серверу. Аргумент команды — пароль |
| PASV | Вход в пассивный режим. С ним мы можем пользоваться такими командами как: LIST, STOR, RETR. Эта команда создает отдельный сокет для передачи файлов по ip и порту, который нам отправит сервер. |
| LIST | После отправки, сервер нам по сокету данных будет отправлять все содержимое текущей директории (файлы, папки). |
| RETR “аргумент” | Скачать файл с сервера. Аргумент команды — имя файла. Скачивание и запись происходит в отдельном потоке. |
| STOR “аргумент” | Загрузить файл на сервер. Аргумент команды — имя файла. Загрузка происходит в отдельном потоке. |
| ABOR | Прервать передачу файла. |
| CWD “аргумент” | Перейти в директорию. Аргумент команды — имя директории. |
| MKD “аргумент” | Создать директорию в текущей директории. Аргумент команды — имя создаваемой директории. |
| PWD | Вернуть имя текущей директории в данный момент. |
| RMD “аргумент” | Удалить директорию. Аргумент команды — имя удаляемой директории. |
| TYPE “аргумент” | Установка режима записи файлов. Аргумент команды — режим(‘I’, ‘A’). ‘A’ — текстовый режим записи. ‘I’ — бинарный режим записи. |
| DELE “аргумент” | Удалить файл. Аргумент команды — имя файла. |
| SIZE “аргумент” | Вернуть размер файла. Аргумент команды — имя файла. |
| QUIT | Отключиться от сервера и завершить работу. |

# **ТЕСТИРОВАНИЕ**

Для тестирования подключимся к серверу и произведем ряд команд.

Для начала нужно “залогиниться” на сервере, введем следующие команды:

|  |  |  |
| --- | --- | --- |
| Команда | Ожидаемый результат | Фактический результат |
| USER Alexei | Сервер скажет чтобы мы дальше вводили пароль. | Успешно. |
| PASS leh123 | Сервер вышлет сообщение о неверном пароле. | Успешно. |
| PASS leha123 | Сервер ответит об успешном логине. | Успешно. |

# **ЗАКЛЮЧЕНИЕ**

В результате работы над этим проектом был разработан FTP-Клиент. Был реализован некоторый список команд для его работы с сервером. Были выполнены все условия, поставленные темой курсового проекта. Клиент был протестирован на сервере. В результате тестирования проблем выявлено не было. Был получен опыт в конструировании ПО.

# **СПИСОК ИСТОЧНИКОВ**

**ПРИЛОЖЕНИЕ А**

**(справочное)**

**Листинг программы**