**Введение**

Файловый менеджер — это инструмент, предназначенный для упрощения управления файлами и папками на вашем устройстве. Он позволяет пользователю выполнять повседневные операции с файлами (такие как копирование, перемещение, удаление, поиск и сортировка) с минимальными усилиями и максимальной эффективностью.

Современные файловые системы могут содержать десятки тысяч файлов, и управлять ими вручную становится всё сложнее. Наш файловый менеджер разработан, чтобы устранить эту проблему, обеспечивая удобный и интуитивно понятный интерфейс для взаимодействия с вашими данными.

**Основные особенности файлового менеджера**

Файловый менеджер обладает следующими ключевыми функциями и возможностями:

* **Копирование и перемещение файлов**: Легко копируйте или перемещайте файлы между разными папками.
* **Создание и удаление папок**: Создавайте новые папки для упорядочивания данных или удаляйте ненужные.
* **Поиск файлов**: Быстрый поиск файлов по имени, дате создания или типу.
* **Работа с архивами**: Создание, распаковка и просмотр содержимого архивов.
* **Управление большими объёмами данных**: Программа оптимизирована для работы с большими директориями.
* **Простота использования**: Удобный интерфейс с интуитивными командами или графическим представлением.

**Кому полезен этот инструмент?**

Наш файловый менеджер предназначен для пользователей с различным уровнем подготовки:

* **Для рядовых пользователей**: Программа поможет упростить ежедневные задачи, такие как поиск и сортировка документов, управление медиафайлами или очистка системы от ненужных файлов.
* **Для разработчиков**: Быстрый доступ к проектным файлам и работа с архивами в одном месте.
* **Для системных администраторов**: Удобство при выполнении массовых операций, таких как резервное копирование данных или организация файловой структуры.

**Почему стоит выбрать наш файловый менеджер?**

1. **Эффективность работы**: Ускоряет выполнение рутинных операций с файлами.
2. **Простота и доступность**: Подходит как для новичков, так и для опытных пользователей.
3. **Надёжность**: Сохраняет целостность данных при любых операциях.
4. **Расширяемость**: Код легко модифицировать для добавления новых функций.

В дальнейшем разделы документации подробно опишут установку программы, её использование, а также предоставят советы по её настройке под ваши потребности.

**Установка**

Для корректной работы файлового менеджера необходимо выполнить несколько шагов по установке. Данная инструкция описывает процесс установки программы на вашу систему.

**Требования**

Перед установкой убедитесь, что на вашем устройстве выполнены следующие условия:

* **Операционная система**: Windows, macOS или Linux.
* **Версия Python**: 3.8 или выше.
* **Свободное место на диске**: не менее 50 МБ.
* **Установленный Git** (опционально, если вы клонируете репозиторий).

**Шаг 1: Загрузка файлов программы**

**Вариант 1: Скачивание через Git**

Если Git установлен на вашем устройстве:

1. Откройте терминал (командную строку).
2. Выполните команду:

bash

git clone https://github.com/ваш-репозиторий/file-manager.git

1. Перейдите в директорию программы:

bash

cd file-manager

**Вариант 2: Скачивание архива**

1. Перейдите на [страницу репозитория](https://github.com/%D0%B2%D0%B0%D1%88-%D1%80%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%B9/file-manager).
2. Нажмите кнопку Code → Download ZIP.
3. Распакуйте скачанный архив в удобное место на вашем устройстве.

**Шаг 2: Установка зависимостей**

1. Убедитесь, что Python установлен и добавлен в PATH. Чтобы проверить это, выполните в терминале:

css

python --version

Если версия Python отображается, можно продолжить.

1. Установите менеджер пакетов pip (если он ещё не установлен). Чтобы проверить:

css

pip --version

Если pip отсутствует, установите его, следуя [официальной инструкции](https://pip.pypa.io/en/stable/installation/).

1. Установите зависимости программы:

pip install -r requirements.txt

**Шаг 3: Запуск программы**

1. Перейдите в директорию с программой (если ещё не сделали этого):

bash

cd path/to/file-manager

1. Запустите главный скрипт:

css

python main.py

1. Если программа успешно запустилась, вы увидите интерфейс файлового менеджера (текстовый или графический, в зависимости от реализации).

**Шаг 4: Настройка (опционально)**

При первом запуске вы можете настроить программу под свои нужды. Например:

* Укажите рабочую директорию по умолчанию.
* Настройте горячие клавиши или параметры интерфейса.

Настройки обычно сохраняются в файле конфигурации (например, config.json), который находится в корневой папке программы.

**Возможные проблемы и их решение**

**Проблема: Python не найден**

**Решение**: Убедитесь, что Python установлен и добавлен в PATH. Установите Python с [официального сайта](https://www.python.org/).

**Проблема: Ошибка установки зависимостей**

**Решение**: Проверьте, что у вас есть доступ в интернет. Убедитесь, что файл requirements.txt находится в директории программы.

**Проблема: Программа не запускается**

**Решение**: Проверьте, правильно ли установлены зависимости, и используете ли вы команду python main.py в директории программы.

На этом установка завершена! Теперь вы можете начать использовать файловый менеджер для управления своими файлами. Если у вас возникнут вопросы, обратитесь к разделу **FAQ** или свяжитесь с разработчиком.

**Основные функции**

Файловый менеджер предоставляет удобные инструменты для работы с файлами и папками. Его функционал покрывает основные операции, которые могут понадобиться пользователю для управления данными на компьютере. Ниже приведён подробный обзор всех основных функций программы.

**1. Копирование файлов и папок**

Позволяет пользователю копировать файлы и папки из одного местоположения в другое. Множественное копирование не поддерживается. Копируется тот файл по которому, непосредственно, было совершено нажатие курсором.

* **Описание**: Копирование сохраняет исходный файл или папку в указанной директории.
* **Как использовать**:
  1. Выберите файл или папку для копирования.
  2. Укажите путь назначения.
  3. Выполните команду copy.

Copy основан на команде copy в cmd для копирования файлов и xcopy для копирования каталогов со всем содержимым.

xcopy выполняется с флагами:

/y - отключен запрос на подтверждение

/s - копирование всех непустых подкаталогов

/е - копирование пустых каталогов

copy выполняется с флагами:

/y - отключен запрос на подтверждение, если в целевой папке есть файл с таким именем, он перезапишется (не баг, а особенность).

**2. Перемещение файлов и папок**

Перемещает файл или папку из одной директории в другую.

* **Описание**: Перемещение удаляет файл из исходного местоположения и переносит его в указанную директорию.
* **Как использовать**:
  1. Выберите файл или папку.
  2. Укажите путь назначения.
  3. Выполните команду move.

**3. Удаление файлов и папок**

Удаляет ненужные файлы или папки из системы.

* **Описание**: Удаление позволяет освободить место на диске, удаляя выбранные файлы или папки.
* **Как использовать**:
  1. Укажите файл или папку для удаления.
  2. Выполните команду delete.
  3. Подтвердите действие (если требуется).

delete основан на базе команды del в cmd. С флагами:

/y - отключен запрос на подтверждение

/s - удаляет файлы из каталога и подкаталогов.

**Примечание**: Удалённые файлы не могут быть восстановлены, если не предусмотрен механизм корзины.

**4. Создание папок**

Позволяет создавать новые директории для организации файлов.

* **Описание**: Создание новых папок помогает упорядочить файлы и облегчает управление данными.
* **Как использовать**:
  1. Укажите путь, где нужно создать папку.
  2. Выполните команду create с указанием имени папки.

## 5. ****Поиск файлов****

Осуществляет поиск файлов по указанным критериям (имя, расширение, дата создания и др.).

* **Описание**: Поиск позволяет быстро находить файлы в заданной директории или по всей системе.
* **Как использовать**:
  1. Укажите директорию для поиска.
  2. Задайте критерии поиска.
  3. Выполните команду search.

## 6. ****Работа с архивами****

Поддерживает создание и извлечение архивов.

**Создание архивов**

* **Описание**: Сжатие файлов или папок в единый архив для экономии места и удобства передачи.
* **Как использовать**:
  1. Укажите файлы или папки для архивации.
  2. Выполните команду archive create.

**Распаковка архивов**

* **Описание**: Извлечение содержимого архива в указанную директорию.
* **Как использовать**:
  1. Укажите путь к архиву.
  2. Выполните команду archive extract.

## 7. ****Сортировка файлов****

Позволяет сортировать файлы в папках по имени, размеру, дате создания или типу.

* **Описание**: Сортировка помогает организовать данные для удобного просмотра.
* **Как использовать**:
  1. Укажите директорию.
  2. Выполните команду sort, указав критерий.

## 8. ****Просмотр информации о файле****

Отображает информацию о выбранном файле или папке.

* **Описание**: Просмотр позволяет узнать размер, дату создания, тип файла и другие метаданные.
* **Как использовать**:
  1. Укажите файл или папку.
  2. Выполните команду info.

## 9. ****Подсчёт размера папок****

Рассчитывает общий объём всех файлов в указанной папке.

* **Описание**: Подходит для анализа использования дискового пространства.
* **Как использовать**:
  1. Укажите папку.
  2. Выполните команду size.

## 11. ****Вырезание файлов и папок****

Перемещает выбранные файлы или папки в буфер обмена для последующей вставки в другое место.

* **Описание**: Команда cut позволяет временно сохранить файл или папку в буфере обмена, а затем вставить в другую директорию, что фактически перемещает объект. В отличие от команды copy, cut удаляет исходный файл или папку после вставки.
* **Как использовать**:
  1. Укажите файл или папку для вырезания.
  2. Выполните команду cut.
  3. Перейдите в директорию назначения.
  4. Выполните команду paste для завершения операции.

cut основан на команде move в cmd. Выполняется с флагом /y - перемещение без подтверждения.

## 11. ****Поддержка горячих клавиш (опционально)****

Для ускорения работы с программой могут быть настроены горячие клавиши.

**Пример горячих клавиш**:

* Ctrl+C — копировать файл.
* Ctrl+V — вставить файл.
* Ctrl+D — удалить файл.

Все эти функции делают файловый менеджер мощным инструментом для управления данными, предоставляя пользователю максимум удобства и контроля.

**Руководство пользователя**

Файловый менеджер предназначен для управления файлами и папками. В данном разделе описаны основные операции, которые пользователь может выполнять, с пошаговыми инструкциями и примерами. Руководство охватывает все ключевые функции программы.

**1. Запуск программы**

1. Убедитесь, что все зависимости установлены (см. раздел "Установка").
2. Перейдите в директорию программы.
3. Запустите команду:

сss

python main.py

1. После запуска программа предложит текстовый или графический интерфейс (в зависимости от реализации).

**2. Навигация по папкам**

Для просмотра содержимого папки:

1. Используйте команду ls для отображения содержимого текущей директории:

bash

ls

1. Чтобы перейти в другую папку, используйте команду cd:

bash

cd /path/to/directory

1. Чтобы вернуться на уровень выше, выполните:

bash

cd ..

**3. Копирование файлов и папок**

Чтобы скопировать файл или папку:

1. Введите команду copy, указав исходный файл и путь назначения:

sql

copy /home/user/documents/file.txt /home/user/backup/

1. Если вы копируете папку, укажите флаг -r для рекурсивного копирования:

sql

copy -r /home/user/documents/ /home/user/backup/

**4. Вырезание и вставка файлов (команда cut)**

1. Вырежьте файл или папку с помощью команды cut:

bash

cut /home/user/documents/file.txt

1. Перейдите в директорию назначения:

bash

cd /home/user/backup/

1. Вставьте объект:

bash

paste

**5. Перемещение файлов и папок**

Перемещение аналогично вырезанию и вставке, но выполняется одной командой:

arduino

move /home/user/documents/file.txt /home/user/backup/

**6. Удаление файлов и папок**

1. Чтобы удалить файл, используйте команду delete:

arduino

delete /home/user/temp/file.txt

1. Для удаления папки используйте флаг -r:

arduino

delete -r /home/user/temp/

**Внимание:** Убедитесь, что вы действительно хотите удалить объект, так как восстановить его будет невозможно.

**7. Создание папок**

Чтобы создать новую папку:

arduino

create/home/user/documents/new\_folder

**8. Поиск файлов**

Для поиска файлов по имени или другим критериям:

1. Выполните команду search, указав директорию и критерий поиска:

sql

search /home/user/documents/ --name "report"

1. Для поиска по расширению файлов:

sql

search /home/user/documents/ --extension ".txt"

**9. Работа с архивами**

**Создание архива**

Для создания архива укажите файлы или папки и путь для сохранения архива:

sql

archive create /home/user/documents/ /home/user/documents.zip

**Распаковка архива**

Для извлечения содержимого архива:

arduino

archive extract /home/user/documents.zip /home/user/extracted/

**10. Сортировка файлов**

Сортировка файлов в папке по дате, размеру или имени:

bash

sort /home/user/documents/ --by date

**11. Просмотр информации о файле**

Чтобы отобразить подробную информацию о файле или папке:

arduino

info /home/user/documents/file.txt

**12. Подсчёт размера папки**

Для определения общего размера содержимого папки:

arduino

size /home/user/documents/

**13. Горячие клавиши (опционально)**

Если ваш файловый менеджер поддерживает горячие клавиши, настройте их через файл конфигурации (например, config.json).

Примеры стандартных сочетаний:

* Ctrl+C — копировать.
* Ctrl+X — вырезать.
* Ctrl+V — вставить.
* Delete — удалить файл.

Руководство пользователя охватывает основные операции, необходимые для эффективного управления файлами и папками. Если у вас возникнут трудности, обратитесь к разделу **FAQ**.

**FAQ: Часто задаваемые вопросы**

В этом разделе собраны наиболее распространённые вопросы, возникающие у пользователей файлового менеджера, и подробные ответы на них.

**1. Программа не запускается. Что делать?**

**Возможные причины:**

* Python не установлен или не добавлен в PATH.
* Зависимости не установлены.

**Решение:**

1. Проверьте, установлен ли Python, выполнив команду:

css

python --version

Если Python отсутствует, скачайте его с [официального сайта](https://www.python.org/) и установите.

1. Убедитесь, что зависимости установлены:

pip install -r requirements.txt

**2. Почему я получаю ошибку при выполнении операций?**

**Возможные причины:**

* У вас недостаточно прав для выполнения операции в указанной директории.
* Вы пытаетесь работать с системными файлами.

**Решение:**

1. Убедитесь, что у вас есть права доступа к файлам или папкам.
2. Попробуйте запустить программу с повышенными правами (например, от имени администратора).

**3. Как восстановить случайно удалённый файл?**

Файловый менеджер не имеет функции корзины, поэтому восстановление удалённых файлов невозможно.  
**Рекомендация:** Используйте сторонние инструменты восстановления данных, такие как Recuva или TestDisk.

**4. Можно ли использовать файловый менеджер для работы с сетевыми хранилищами?**

На данный момент поддержка работы с сетевыми дисками или облачными хранилищами не реализована. Однако вы можете подключить сетевые ресурсы как локальные диски (например, через smb или nfs), после чего файловый менеджер сможет работать с ними как с обычными директориями.

**5. Как изменить язык интерфейса программы?**

Если файловый менеджер поддерживает многоязычность, настройка языка выполняется через файл конфигурации (например, config.json).

**Пример изменения языка:**

1. Откройте файл config.json в текстовом редакторе.
2. Найдите параметр language и измените его значение, например:

json

{

"language": "en"

}

1. Сохраните изменения и перезапустите программу.

**6. Как отключить подтверждение перед удалением файлов?**

Подтверждение удаления предотвращает случайные ошибки, но вы можете отключить его через настройки.

**Шаги:**

1. Откройте файл config.json.
2. Найдите параметр confirm\_delete и установите его значение в false:

json

{

"confirm\_delete": false

}

1. Сохраните изменения.

**7. Программа выдает ошибку при работе с большими папками. Что делать?**

**Возможные причины:**

* Недостаточно оперативной памяти.
* Ограничение файловой системы.

**Решение:**

1. Разделите содержимое больших папок на несколько меньших.
2. Попробуйте увеличить объём доступной памяти, закрыв ненужные программы.
3. Проверьте, поддерживает ли ваша файловая система количество файлов в папке (например, FAT32 имеет ограничения).

**8. Можно ли настраивать горячие клавиши?**

Да, горячие клавиши настраиваются в файле конфигурации (config.json).

**Пример:**

1. Откройте файл config.json.
2. Добавьте или измените секцию с горячими клавишами, например:

json

{

"hotkeys": {

"copy": "Ctrl+C",

"paste": "Ctrl+V",

"delete": "Del"

}

}

1. Сохраните файл и перезапустите программу.

**9. Почему программа не видит определённые файлы?**

**Возможные причины:**

* Файлы скрыты в системе.
* Указанная директория недоступна.

**Решение:**

1. Убедитесь, что файлы не скрыты. Для их отображения проверьте настройки вашей операционной системы.
2. Проверьте права доступа к директории.

**10. Могу ли я добавить свои команды в файловый менеджер?**

Да, файловый менеджер поддерживает расширяемость. Вы можете добавить свои команды, изменив исходный код.

**Шаги:**

1. Откройте файл с основным кодом программы (например, main.py).
2. Найдите секцию, где обрабатываются команды.
3. Добавьте свою команду, следуя структуре существующего кода.
4. Сохраните изменения и перезапустите программу.

**11. Как восстановить настройки по умолчанию?**

Удалите файл конфигурации config.json. При следующем запуске программы будет создан новый файл с настройками по умолчанию.

**12. Программа работает слишком медленно. Что делать?**

**Возможные причины:**

* Слишком большое количество файлов в обрабатываемой директории.
* Нехватка ресурсов системы.

**Решение:**

1. Попробуйте уменьшить количество файлов в папке, которую вы обрабатываете.
2. Закройте фоновые процессы, чтобы освободить ресурсы.
3. Если проблема сохраняется, обратитесь к разработчику.

**13. Где можно сообщить об ошибке или предложить улучшение?**

Вы можете сообщить об ошибке или оставить предложение в разделе "Issues" на странице [репозитория программы](https://github.com/%D0%B2%D0%B0%D1%88-%D1%80%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%B9/file-manager).

Если вы не нашли ответа на свой вопрос, свяжитесь с разработчиком или ознакомьтесь с дополнительной документацией.

**Лицензия и информация об авторе**

**Условия использования:**

1. **Свобода использования**: Вы можете использовать, копировать, изменять и распространять данный программный продукт как в коммерческих, так и в некоммерческих целях.
2. **Отсутствие гарантий**: Программа предоставляется "как есть" без каких-либо явных или подразумеваемых гарантий, включая гарантии товарной пригодности или соответствия определённой цели.
3. **Кредит авторства**: При использовании исходного кода необходимо сохранить указание на авторство, включённое в исходные файлы.

**Информация об авторе**

**Автор/Команда разработчиков:**

* **Имя автора**: [Ваше имя или псевдоним]
* **Контактный email**: [email@example.com]
* **GitHub**: [https://github.com/ваш-аккаунт](https://github.com/%D0%B2%D0%B0%D1%88-%D0%B0%D0%BA%D0%BA%D0%B0%D1%83%D0%BD%D1%82)
* **Веб-сайт/Блог**: https://ваш-сайт.example.com

**Благодарности:**

* Спасибо всем, кто помогал тестировать и улучшать данный проект.
* Особая благодарность [указать имена или команды, которые оказали значимую поддержку].

**Вклад и поддержка**

Если вы хотите внести вклад в развитие файлового менеджера:

1. Форкните репозиторий на [GitHub](https://github.com/%D0%B2%D0%B0%D1%88-%D1%80%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%B9).
2. Внесите свои изменения.
3. Создайте pull request с описанием ваших улучшений.

Для отчётов об ошибках и предложений обратитесь к разделу "Issues" на странице репозитория.

**Поддержка**: Если у вас есть вопросы или предложения, свяжитесь с автором по указанным контактам.