

Презентация на тему: *Linux terminal (part 1)*

Подготовлено @kotlena01 для команды #Android
18.03.2025



Linux терминал для Windows — это программное обеспечение, которое позволяет пользователям Windows запускать команды, аналогичные тем, что используются в операционной системе Linux. В Windows по умолчанию нет встроенной поддержки командной строки, которая бы поддерживала команды Linux, поэтому для работы с такими командами нужны дополнительные инструменты.

Git Bash — это одно из таких приложений, которое предоставляет пользователям Windows эмулятор Linux-терминала, поддерживающий команды и утилиты, часто используемые в Linux. Git Bash является частью установки Git для Windows и включает в себя множество командных утилит, которые позволяют работать с Git и выполнять другие операции через командную строку, как в Linux.

Для чего нужны Linux терминал и Git Bash?

1. Разработка программного обеспечения: Git Bash позволяет разработчикам на Windows использовать команды Git, что важно для работы с системами управления версиями. Это особенно полезно для разработчиков, работающих с открытым исходным кодом или командных проектов, где важна интеграция с Git.
2. Совместимость с Linux-средой: Многие инструменты и фреймворки, такие как Docker, Python или Node.js, активно используют команды Linux. Git Bash даёт возможность запускать эти команды на Windows, что делает разработку проще и позволяет избежать проблем с несовместимостью.
3. Удобство для разработчиков: Разработчики, которые уже привыкли к работе с терминалом Linux или macOS, могут использовать Git Bash, чтобы использовать знакомые команды и утилиты на Windows. Это ускоряет рабочий процесс и помогает избежать необходимости переходить между операционными системами.
4. Автоматизация задач и скрипты: Git Bash предоставляет возможность выполнять сложные операции через командную строку, например, создавать скрипты для автоматизации различных задач. Это полезно для настройки и управления проектами, работы с базами данных или системами на сервере.
5. Мощные инструменты для работы с файлами и папками: В отличие от стандартной командной строки Windows, Git Bash поддерживает команды Linux, такие как `ls`, `cp`, `mv`, `rm`, что позволяет более гибко управлять файлами и папками.

Git Bash и Linux терминал для Windows необходимы для пользователей, которые хотят работать в среде, подобной Linux, на операционной системе Windows. Эти инструменты особенно полезны для разработчиков программного обеспечения, работающих с Git и системами на основе Linux, а также для автоматизации задач и использования командной строки для выполнения операций. Git Bash помогает создать знакомую рабочую среду для пользователей, которые хотят избежать перехода на Linux или macOS, сохраняя при этом все преимущества командной строки.

Для выполнения задания по Linux terminal должен быть установлен GitBash для Windows.

Чтобы установить Git Bash для Windows, выполните следующие шаги:

1. Перейдите на официальный сайт Git по адресу git-scm.com (<https://git-scm.com/>).
2. На главной странице нажмите на кнопку "Download" для Windows. Система автоматически определит вашу версию.
3. Скачайте установочный файл, кликнув по ссылке.
4. Запустите загруженный установочный файл.
5. Следуйте указаниям мастера установки:
 - На экране выбора компонентов оставьте все опции по умолчанию или выберите нужные.
 - Выберите редактор текста по умолчанию (например, Vim или Nano).
 - Выберите настройку PATH, чтобы Git был доступен из командной строки.
6. Пройдите через остальные шаги установки, оставляя настройки по умолчанию или изменяя их в зависимости от ваших предпочтений.
7. По окончании установки найдите Git Bash в меню "Пуск" и запустите его.

Для выполнения задания по Linux terminal должен быть создан аккаунт в GitHub.

Чтобы создать аккаунт на сайте GitHub, следуйте этим шагам:

1. Зайдите на сайт <https://github.com/>
2. Нажмите на кнопку "Sign up" в верхнем правом углу страницы.
3. Введите ваш адрес электронной почты и нажмите "Continue".
4. Создайте пароль для вашего аккаунта и нажмите "Continue".
5. Введите ваше имя пользователя. Оно должно быть уникальным. Нажмите "Continue".
6. Выберите план использования (бесплатный или платный).
7. Подтвердите ваш адрес электронной почты, следуя инструкциям в письме, которое вы получите от GitHub.
8. После подтверждения электронной почты вы сможете войти в свой новый аккаунт.

Для выполнения задания по **Linux terminal** должен быть создан локальный репозиторий **Investmate** на компьютере и запущен на **GitHub**.

Чтобы создать локальный репозиторий **Investmate** на компьютере и запустить его на удалённый сервер **GitHub**, следуйте этим шагам:

1. Создайте новую директорию для вашего репозитория: `mkdir Investmate` и переместитесь в нее: `cd Investmate`

2. Инициализируйте репозиторий: `git init`

3. Добавьте файлы в репозиторий:

Создайте или переместите файлы в созданную директорию. Затем добавьте их, с помощью команды `git add .`

4. Сделайте первый коммит: `git commit -m "Первый коммит"`

5. Создайте удалённый репозиторий:

- Перейдите на **GitHub** и создайте новый репозиторий **Investmate**. Обратите внимание на URL вашего нового репозитория.

6. Добавьте удалённый репозиторий: `git remote add origin https://github.com/ваш-логин/имя-репозитория.git`

7. Запушьте изменения на удалённый репозиторий: `git push -u origin master`

Либо созданный удаленный репозиторий на GitHub должен быть скопирован на локальный компьютер в отдельную папку.

Чтобы создать репозиторий на GitHub, выполните следующие шаги:

1. Войдите в свой аккаунт на GitHub.
2. На главной странице нажмите на значок "+" в верхнем правом углу и выберите "New repository".
3. Введите название вашего репозитория в поле **"Investmate"**.
4. (Опционально) Добавьте описание вашего репозитория в поле "Description".
5. Выберите видимость репозитория:
 - Public – репозиторий будет доступен всем.
 - Private – репозиторий будет доступен только вам и тем, кому вы предоставите доступ.
6. (Опционально) Вы можете инициализировать репозиторий с помощью файла README, выбрать лицензию или добавить .gitignore.
7. Нажмите на кнопку "Create repository".

Чтобы клонировать репозиторий с GitHub, выполните следующие шаги.

1. Скопируйте URL репозитория:

- Перейдите на страницу репозитория Investmate на GitHub.
- Нажмите на кнопку "Code" и скопируйте предоставленный URL (выбирайте между HTTPS и SSH).

2. Откройте терминал:

- На вашем компьютере откройте терминал (или командную строку).

3. Используйте команду git clone:

Введите следующую команду, заменив URL на скопированный вами адрес репозитория: `git clone URL`

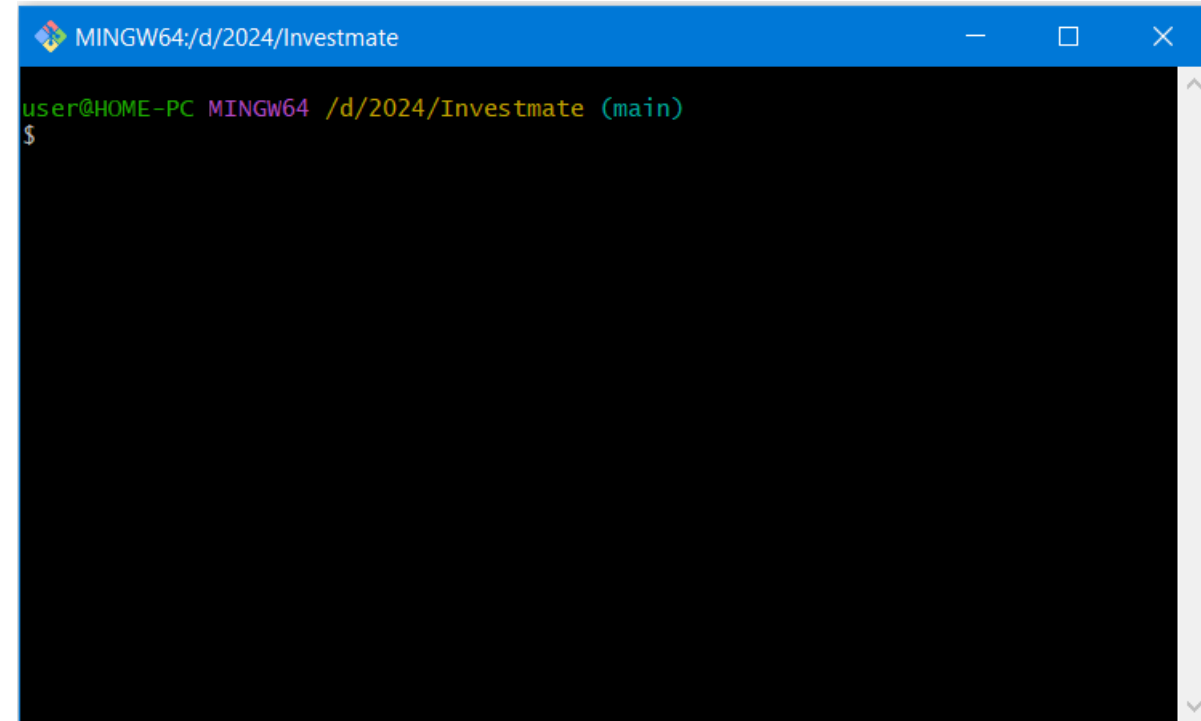
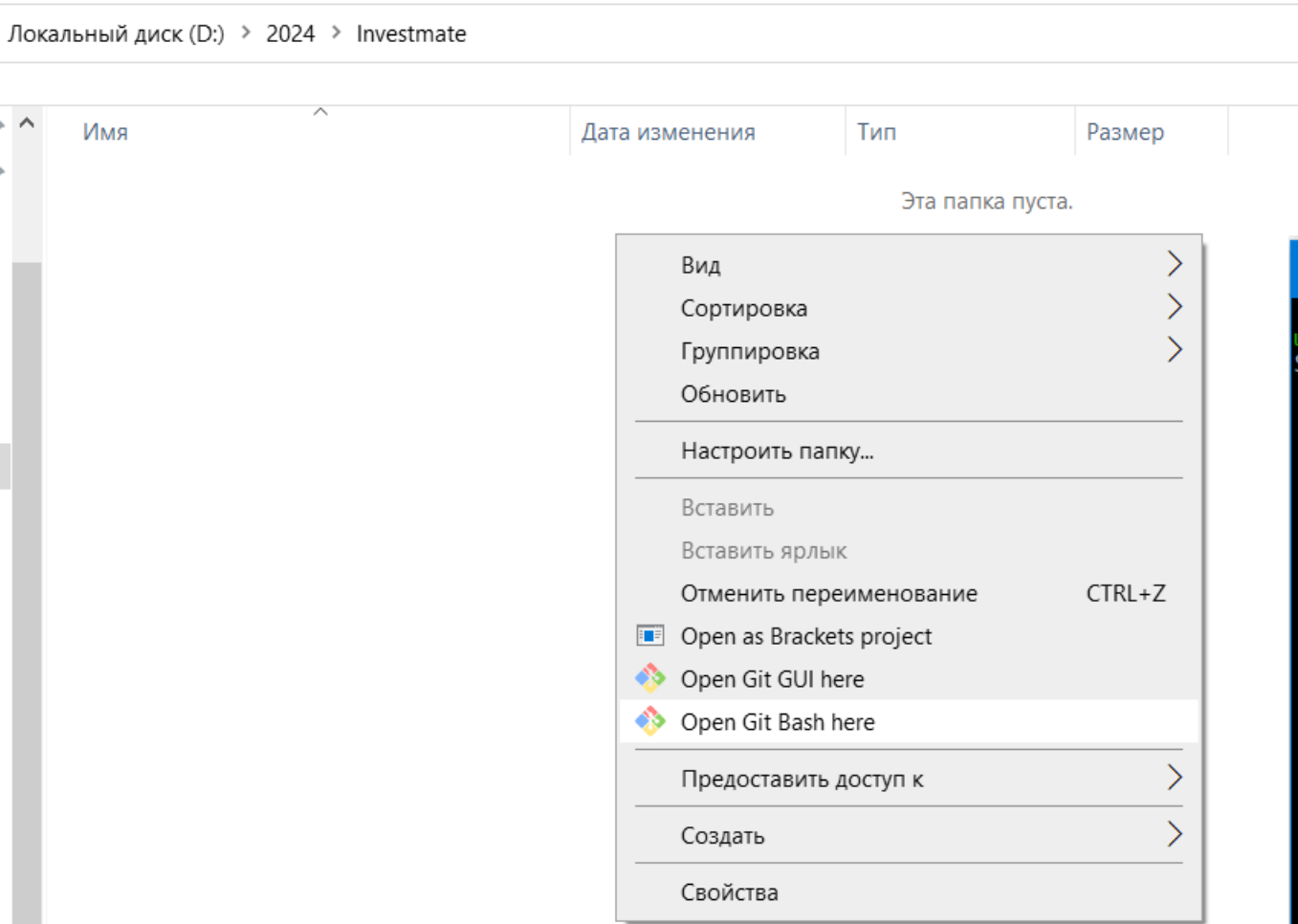
4. Перейдите в папку с репозиторием:

После завершения клонирования вы можете перейти в папку с репозиторием Investmate:

```
cd Investmate
```

Запустите командную строку

Откройте локальный репозиторий Investmate на компьютере. Нажмите правой кнопкой мыши Open Git Bash here, откроется Linux-терминал



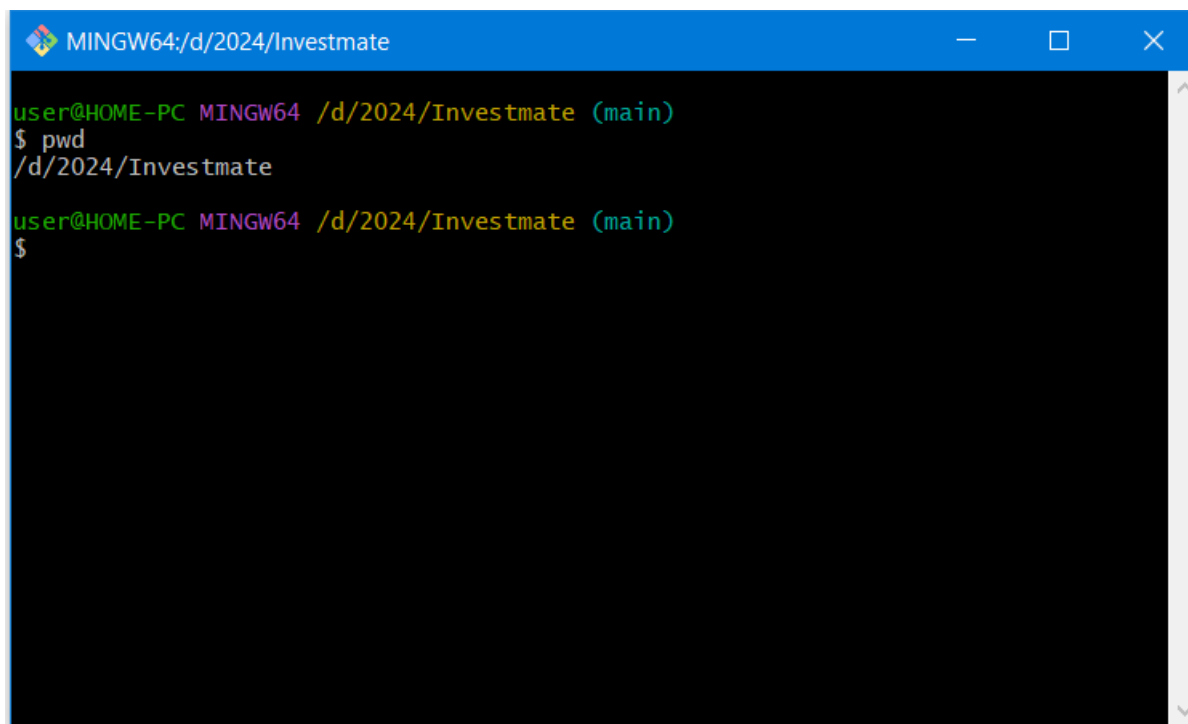
Команда pwd

Введите в терминале команду **pwd**, затем нажмите enter.

Команда pwd используется для отображения текущего рабочего каталога (папки), в котором вы находитесь в данный момент. Она расшифровывается как "print working directory".

Когда вы вводите эту команду в терминале, вы получаете полный путь к каталогу. Это полезно для навигации и ориентации в файловой системе. Запомните эту полезную команду, она особенно пригодится когда у вас нет ничего, кроме командной строки под рукой!

/d/2024/Investmate — отображает, где я сейчас нахожусь.

A screenshot of a MINGW64 terminal window. The title bar is blue and contains the text "MINGW64:/d/2024/Investmate" along with standard window control buttons. The terminal has a black background with green text. It shows the prompt "user@HOME-PC MINGW64 /d/2024/Investmate (main)" followed by the command "\$ pwd" and its output "/d/2024/Investmate". Below this, the prompt is shown again with a dollar sign "\$" indicating it is ready for the next command.

```
user@HOME-PC MINGW64 /d/2024/Investmate (main)
$ pwd
/d/2024/Investmate
user@HOME-PC MINGW64 /d/2024/Investmate (main)
$
```

Команды **mkdir**, **mkdir -p**

Создайте в терминале три папки: Learning, Trade, Settings, используя команду **mkdir**

Вы можете создавать папки по одной, например:

mkdir Learning

А можете сделать быстрее и создать несколько папок одновременно, просто указав их имена через пробел:

mkdir Learning Trade Settings

Создайте вложенную папку YouTube внутри папки Capital, используя команду **mkdir** и флаг **-p**:

mkdir -p Capital/YouTube

Для быстрого создания большого количества директорий (например, 10) с последовательными именами используйте команду **mkdir {1..10}**

При выполнении этой команды будут созданы 10 папок: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

Также можно использовать другие диапазоны или комбинации, например:

mkdir {a,b,c}

Будут созданы папки с именами a, b и c.

Командой **mkdir** можно создать папку не только в текущей директории, но и в абсолютно другом месте, где вы сейчас не находитесь, используя абсолютный путь:

mkdir /path/to/destination/new_folder

mkdir /D/2024/Investmate/new_folder

mkdir -p /path/to/destination/new_folde/subfolder

mkdir -p /D/2024/Investmate/new_folder/subfolder

Команды `ls`, `ls -l`, `ls -la`

Введите в терминале команду `ls`, затем нажмите `enter`

Команда `ls` выводит список файлов и директорий в текущем каталоге, по умолчанию в алфавитном порядке. Не показывает скрытые файлы (те, что начинаются с точки).

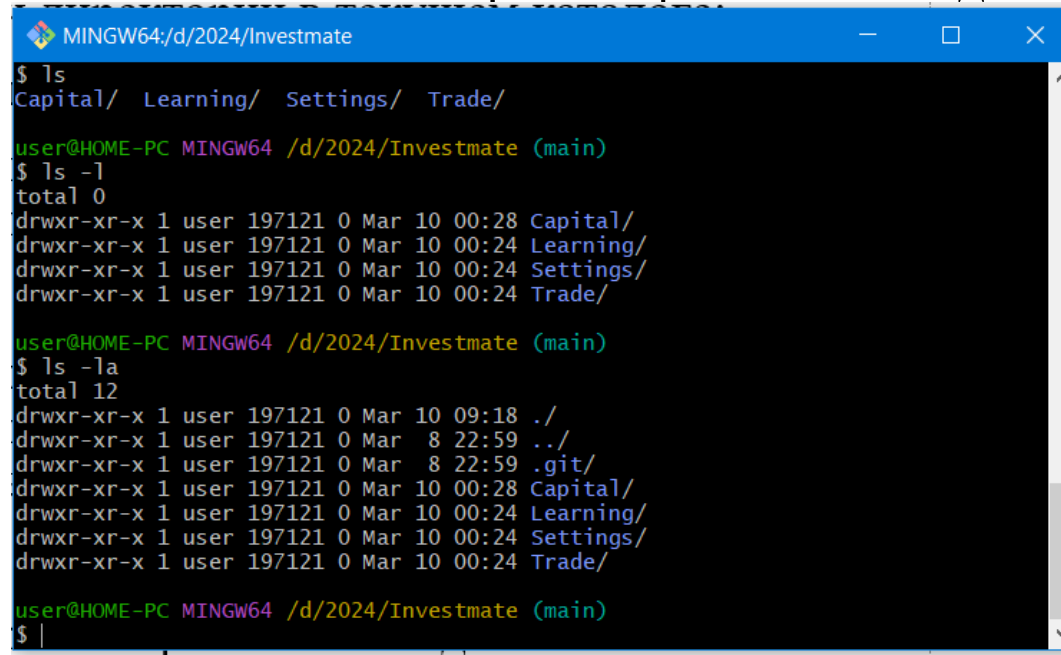
Введите в терминале команду `ls -l`, затем нажмите `enter`

Команда `ls -l` выводит подробный список файлов и директорий, включая информацию о правах доступа, количестве ссылок, владельце, группе, размере и времени последнего изменения. Также не показывает скрытые файлы.

Введите в терминале команду `ls -la`, затем нажмите `enter`

Команда `ls -la` аналогична команде `ls -l`, но также включает скрытые файлы в выводе.

Таким образом команды `ls`, `ls -l`, `ls -la` отработывают в терминале:



```
MINGW64:/d/2024/Investmate
$ ls
Capital/  Learning/  Settings/  Trade/

user@HOME-PC MINGW64 /d/2024/Investmate (main)
$ ls -l
total 0
drwxr-xr-x 1 user 197121 0 Mar 10 00:28 Capital/
drwxr-xr-x 1 user 197121 0 Mar 10 00:24 Learning/
drwxr-xr-x 1 user 197121 0 Mar 10 00:24 Settings/
drwxr-xr-x 1 user 197121 0 Mar 10 00:24 Trade/

user@HOME-PC MINGW64 /d/2024/Investmate (main)
$ ls -la
total 12
drwxr-xr-x 1 user 197121 0 Mar 10 09:18 ./
drwxr-xr-x 1 user 197121 0 Mar 8 22:59 ../
drwxr-xr-x 1 user 197121 0 Mar 8 22:59 .git/
drwxr-xr-x 1 user 197121 0 Mar 10 00:28 Capital/
drwxr-xr-x 1 user 197121 0 Mar 10 00:24 Learning/
drwxr-xr-x 1 user 197121 0 Mar 10 00:24 Settings/
drwxr-xr-x 1 user 197121 0 Mar 10 00:24 Trade/

user@HOME-PC MINGW64 /d/2024/Investmate (main)
$
```

Команды touch, echo, cat, nano, vim (vi) команды для создания файлов

- Команда **touch** создаст пустой файл с указанным именем

Введите в терминале **touch lesson.txt**

- Команда **echo** создаст файл и запишет в него указанный текст

Введите в терминале **echo "Gold, TSLA, AAPL.." > tradinginstruments.json**

- Команда **cat** позволяет создать файл и сразу же ввести в него текст. Для завершения ввода нажмите Ctrl + D

Введите в терминале **cat > language.md**

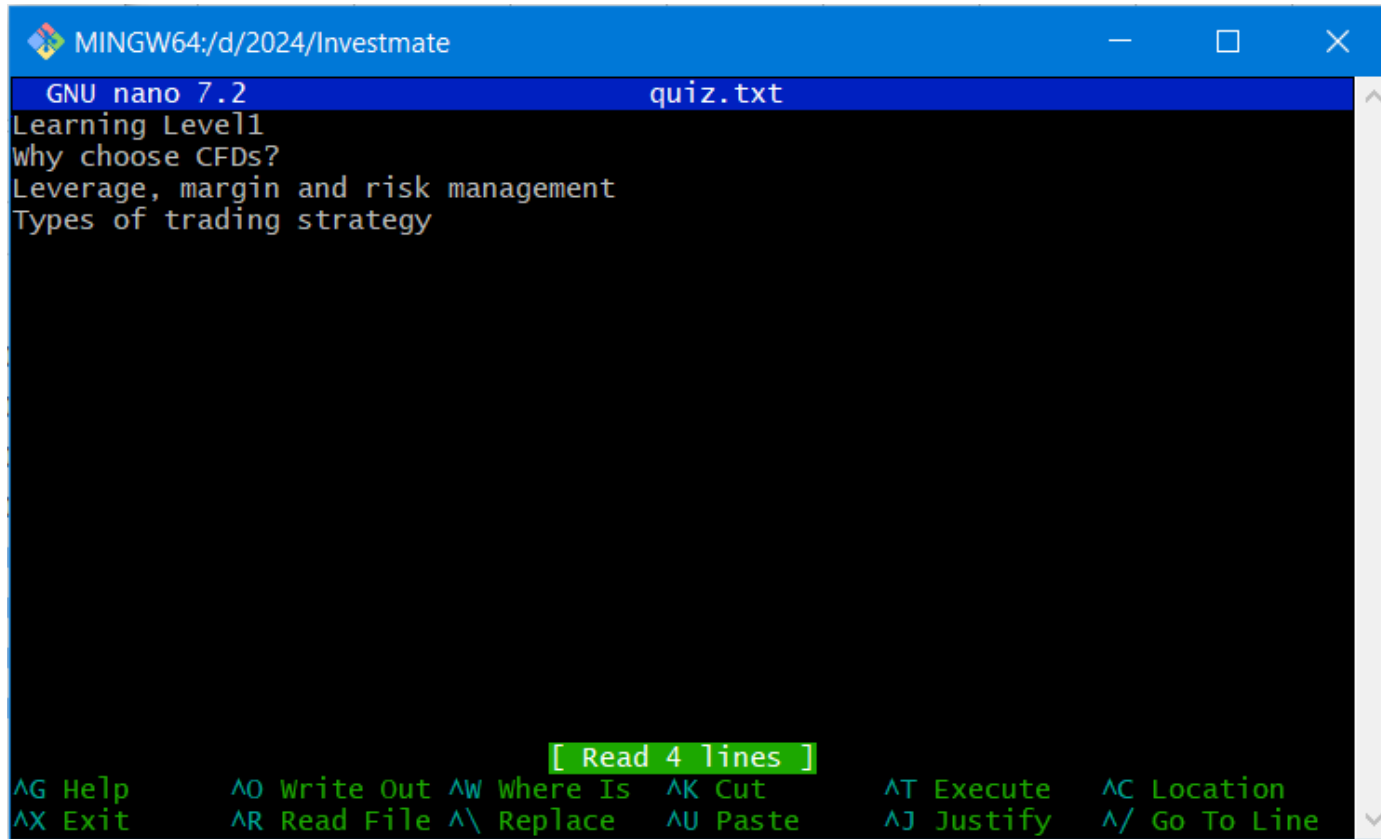
*Редакторы текстов, такие как **nano** или **vim**, вы также можете использовать для создания файлов:*

Введите в терминале : **nano quiz.txt**

Введите любой текст (для перемещения по файлу используйте на клавиатуре кнопки со стрелками), для сохранения изменений нажмите Ctrl + O, затем нажмите Enter. Чтобы выйти из nano, нажмите Ctrl + X.

Приятный бонус в nano – подсказки внизу экрана.

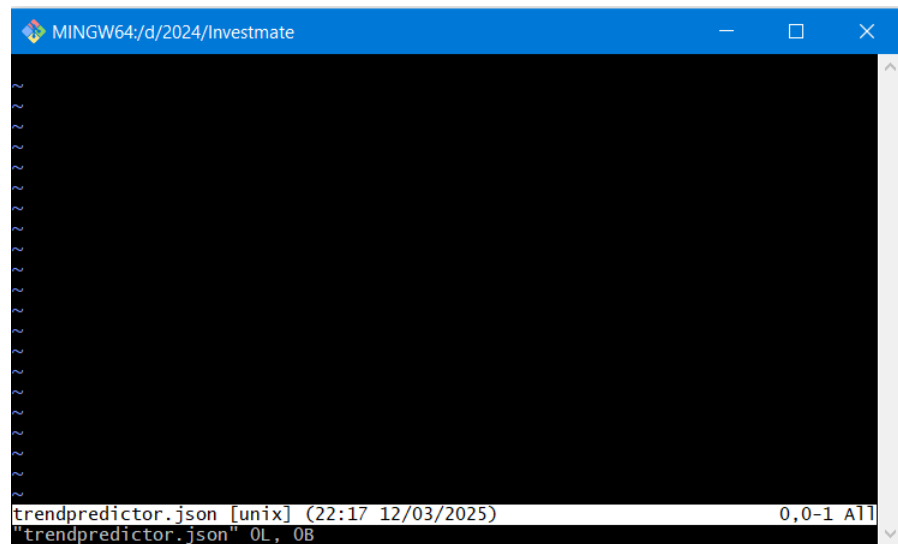
Таким образом команда **nano** отрабатывает в терминале:



```
MINGW64:/d/2024/Investmate
GNU nano 7.2 quiz.txt
Learning Level1
Why choose CFDs?
Leverage, margin and risk management
Types of trading strategy

[ Read 4 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Введите в терминале: **vim trendpredictor.json** или **vi trendpredictor.json** (предшественник vim)
Нажмите i (insert) для перехода в режим вставки и введите текст, который хотите добавить в файл.
Вот так выглядит окно терминала, заблокированное текстовым редактором vim



Как же выйти из Vim???



Чтобы выйти из Vim, можно использовать следующие команды:

1. Выйти, если нет сохраненных изменений:

Нажмите **Esc**, чтобы перейти в командный режим

Введите **:q** и нажмите **Enter**

```
~  
trendpredictor.json [unix] (22:17 12/03/2025) 0,0-1 All  
:q
```

2. Выйти без сохранения изменений:

Нажмите **Esc**, введите **:q!** и нажмите **Enter**

```
~  
trendpredictor.json [unix] (22:17 12/03/2025) 0,0-1 All  
:q!
```

N/B: При изменении в файле, командой **:q** выйти из Vim не возможно, в командной строке необходимо добавить **!**

```
trendpredictor.json[+] [unix] (22:17 12/03/2025) 3,9 All  
E37: No write since last change (add ! to override)
```

3. Сохранить изменения и выйти:

Нажмите **Esc**, введите **:wq** и нажмите **Enter**

```
trendpredictor.json [unix] (22:17 12/03/2025) 0,0-1 All  
:wq
```

Нажмите **Esc**, введите **:x** и нажмите **Enter**

```
trendpredictor.json [unix] (00:48 13/03/2025) 0,0-1 All  
:x
```

Нажмите **Esc**, введите **ZZ** (две большие буквы 'Z')

```
trendpredictor.json [unix] (22:17 12/03/2025) 0,0-1 All  
"trendpredictor.json" 0L, 0B Z
```

Сразу после введения второй буквы **Z**, Vim закроется.

: - сигнализирует о том, что вы собираетесь ввести команду

:q — выйти из редактора (от англ. quit),

! — принудительно выйти без сохранения изменений.

:w — сохранить изменения в текущем файле (от англ. write).

:x - команда для сохранения и выхода

Команды `rmdir`, `rm -r`, `rm`, `mv` для удаления папок (каталогов), файлов, их переименования

Команда **`rmdir directory_name`** удаляет пустые каталоги (remove directory)

Команда **`rm -r directory_name`** удаляет каталог и все содержимое, пример: **`rm -r Trade`**

Команда **`rm -rf directory_name`** для принудительного удаления каталога, даже если в нем есть файлы с атрибутами «только для чтения»

Команда **`rm -ri directory_name`** для удаления каталога с подтверждением. Система будет запрашивать подтверждение удаления для каждого файла в каталоге, вам нужно будет ввести `y` (да) или `n` (нет). Файл останется, при ответе `n` (нет).

Команда **`rm -rfv directory_name`** для принудительного удаления каталога, выведет имена удаляемых файлов

Команда для удаления нескольких файлов: **`rm file1 file2 file3`**

Команда **`rm`** для удаления одного файла: **`rm filename`**, пример: **`rm quiz.txt`**

Команда **`rm -f filename`** для принудительного удаления файла, даже если в нем есть файлы с атрибутами «только для чтения».

Команда **`rm -i filename.txt`** для удаления файла с подтверждением

Удаление всех файлов в каталоге: **`rm *`** # Удаляет все файлы в текущем каталоге

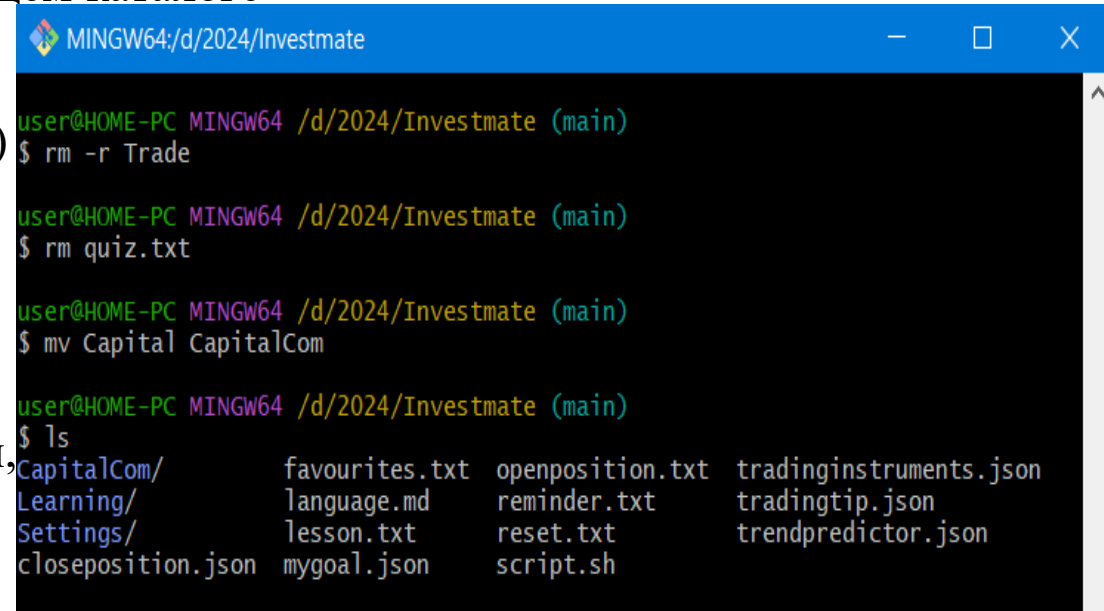
Переименование файла или папки: **`mv old_name new_name`**

пример: **`mv Capital CapitalCom`**

- **`f`** принудительное удаление (игнорирует предупреждения, ошибки)
- **`r`** рекурсивное удаление для (каталогов)
- **`i`** интерактивный режим (спрашивает подтверждение удаления)

Флаг `i` используется для предотвращения случайного удаления данных. Это особенно полезно в случае с важными файлами и каталогами, содержащими критическую информацию.

- **`v`** выводит подробную информацию о действиях (например, файлы, которые удаляются)



```
MINGW64; d/2024/Investmate

user@HOME-PC MINGW64 /d/2024/Investmate (main)
$ rm -r Trade

user@HOME-PC MINGW64 /d/2024/Investmate (main)
$ rm quiz.txt

user@HOME-PC MINGW64 /d/2024/Investmate (main)
$ mv Capital CapitalCom

user@HOME-PC MINGW64 /d/2024/Investmate (main)
$ ls
CapitalCom/      favourites.txt  openposition.txt  tradinginstruments.json
Learning/        language.md    reminder.txt       tradingtip.json
Settings/        lesson.txt     reset.txt          trendpredictor.json
closeposition.json  mygoal.json    script.sh
```


Создайте в Linux-терминале еще 8 файлов, используя удобные вам, вышеперечисленные команды со следующими именами: tradingtip, reset, openposition, closeposition, reminder, mygoal, favourites, script.

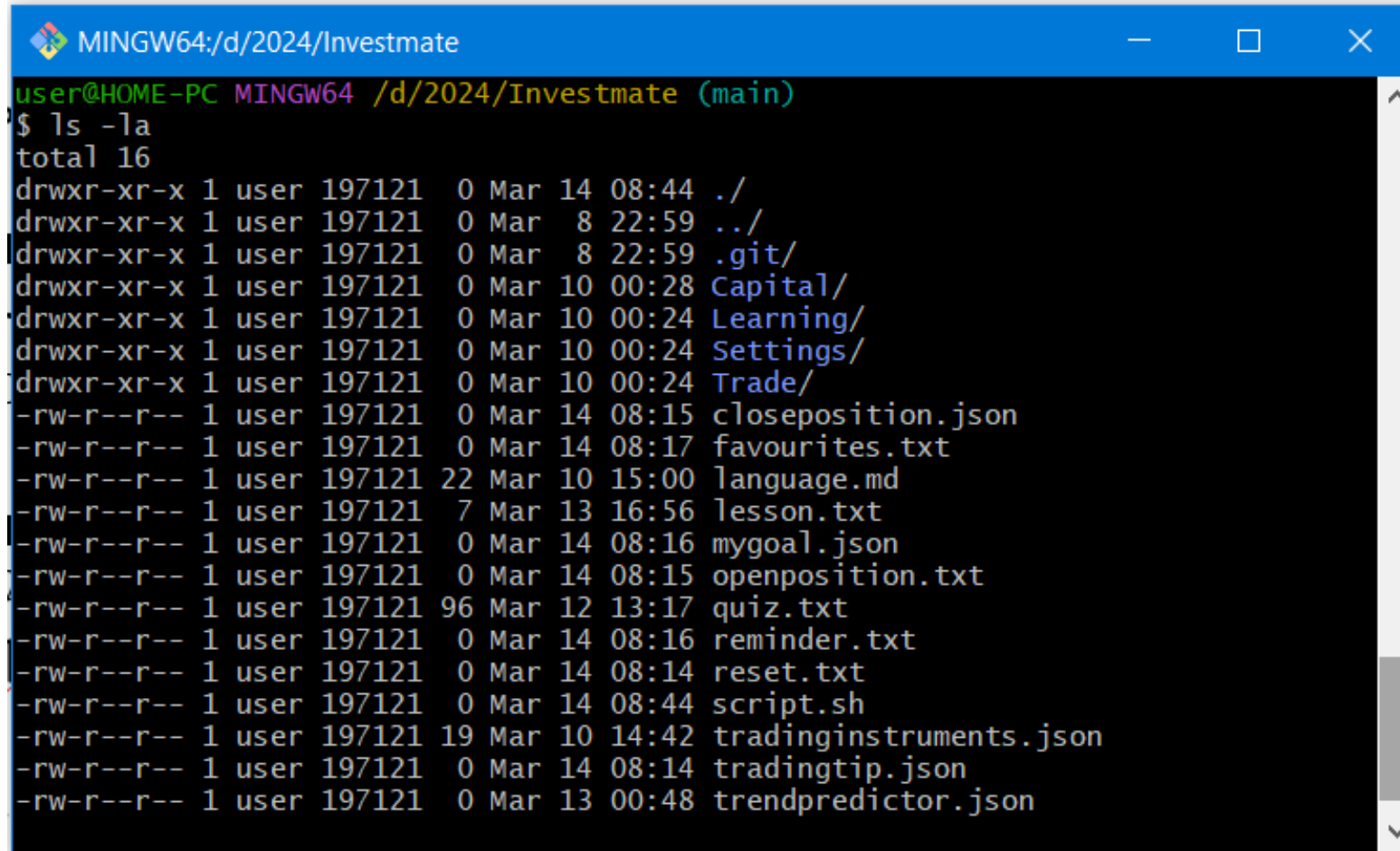
Для информации:

В терминале Linux можно создавать файлы различных форматов. Вот некоторые из них:

1. Текстовые файлы: .txt простой текст, .md файлы Markdown, .csv файлы с разделителями запятыми (обычно для таблиц), .log файлы журналов
2. Файлы конфигурации: .conf файлы конфигурации приложений, .ini файлы INI для настройки программ
3. Файлы данных: .json файлы данных в формате JSON, .xml файлы данных в формате XML, .yaml или .yml файлы YAML для конфигурации
4. Скрипты: .sh скрипты оболочки (bash), .py скрипты Python, .rb скрипты Ruby, .pl скрипты Perl
5. Данные для программ: .sql SQL-скрипты (для работы с базами данных), .html файлы HTML для веб-страниц
6. Архивы: .zip, .tar, .gz файлы архивов различных форматов
7. Программные файлы: .exe исполняемые файлы (обычно для Windows, но могут пересекаться), .bin бинарные файлы

Вы можете создавать файлы любого из этих форматов с помощью команд, упомянутых ранее, изменяя только расширение файла.

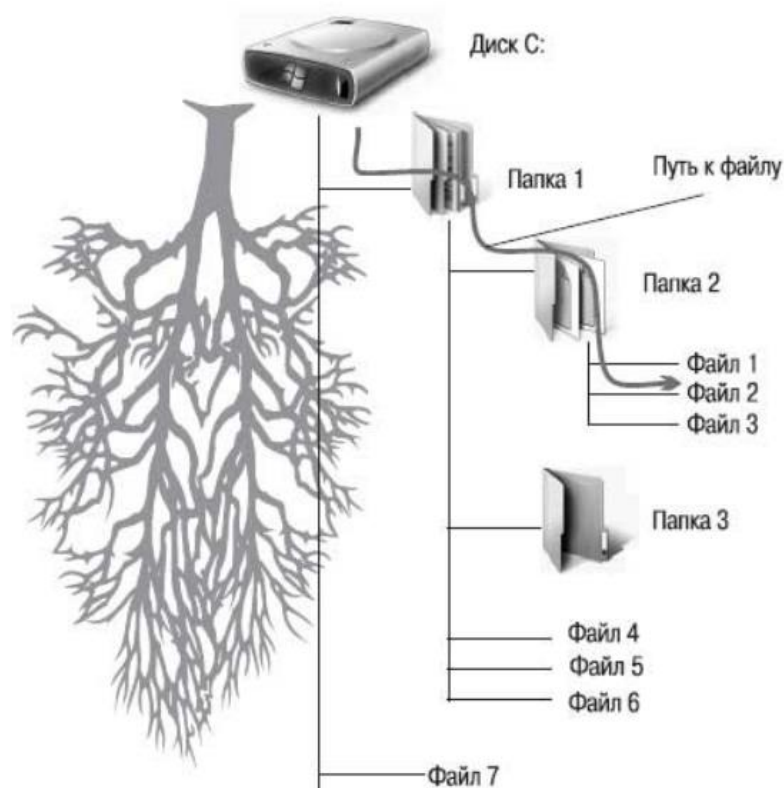
Введите команду `ls -la` и выведите подробный список содержимого папки Investmate:



```
MINGW64:/d/2024/Investmate
user@HOME-PC MINGW64 /d/2024/Investmate (main)
$ ls -la
total 16
drwxr-xr-x 1 user 197121  0 Mar 14 08:44 ./
drwxr-xr-x 1 user 197121  0 Mar  8 22:59 ../
drwxr-xr-x 1 user 197121  0 Mar  8 22:59 .git/
drwxr-xr-x 1 user 197121  0 Mar 10 00:28 Capital/
drwxr-xr-x 1 user 197121  0 Mar 10 00:24 Learning/
drwxr-xr-x 1 user 197121  0 Mar 10 00:24 Settings/
drwxr-xr-x 1 user 197121  0 Mar 10 00:24 Trade/
-rw-r--r-- 1 user 197121  0 Mar 14 08:15 closeposition.json
-rw-r--r-- 1 user 197121  0 Mar 14 08:17 favourites.txt
-rw-r--r-- 1 user 197121 22 Mar 10 15:00 language.md
-rw-r--r-- 1 user 197121  7 Mar 13 16:56 lesson.txt
-rw-r--r-- 1 user 197121  0 Mar 14 08:16 mygoal.json
-rw-r--r-- 1 user 197121  0 Mar 14 08:15 openposition.txt
-rw-r--r-- 1 user 197121 96 Mar 12 13:17 quiz.txt
-rw-r--r-- 1 user 197121  0 Mar 14 08:16 reminder.txt
-rw-r--r-- 1 user 197121  0 Mar 14 08:14 reset.txt
-rw-r--r-- 1 user 197121  0 Mar 14 08:44 script.sh
-rw-r--r-- 1 user 197121 19 Mar 10 14:42 tradinginstruments.json
-rw-r--r-- 1 user 197121  0 Mar 14 08:14 tradingtip.json
-rw-r--r-- 1 user 197121  0 Mar 13 00:48 trendpredictor.json
```

Древовидная файловая система — это структура, где файлы и каталоги (папки) организованы в виде дерева, состоящего из узлов. В этой системе каждый элемент (папка или файл) имеет свою роль и расположение относительно других элементов.

Корневая папка — это верхний уровень файловой системы, от которого все остальные каталоги и файлы расходятся. В Linux корневая папка обозначается символом **/**. Это начальная точка пути, и все другие каталоги и файлы находятся под ней.



Родительская папка — это каталог, который содержит другой каталог или файл. В файловой системе Linux для обозначения родительского каталога используется специальный символ **..** (две точки).

Текущий каталог — это каталог, в котором вы находитесь в данный момент в своей файловой системе, это конкретная папка, из которой был запущен процесс, и с которой он работает. Текущий каталог является основой для выполнения команд и работы с файлами. Команда **pwd** выводит полный путь к вашему текущему каталогу.

Рабочий каталог — это каталог, в котором процесс ожидает выполнять операции, такие как чтение или запись файлов. Для пользователя в командной строке рабочий каталог и текущий каталог совпадают, поскольку текущий каталог — это та директория, в которой вы находитесь, и это рабочий каталог вашей сессии. Однако для программ или процессов, когда они выполняются, их рабочий каталог может быть другим, отличным от того, где находится пользователь. Например, если программа открывает файл, она делает это в своем рабочем каталоге.

Рабочий каталог может быть изменен с помощью команды **cd**, которая меняет рабочий каталог на указанный вами. Для обозначения текущего каталога используется символ **.** (одна точка).

Пример **cd ./Settings**. Здесь точка обозначает текущий каталог, и команда переместит вас в каталог Settings внутри текущего каталога Investmate.

Операционная система осуществляет доступ к файлу или каталогу через его путевое имя или, попросту путь. Существуют два вида путей, абсолютный и относительный.

Абсолютный путь — это полный путь от корневой папки до указанного каталога или файла. Абсолютные пути всегда начинаются с символа **/**, который обозначает корень файловой системы.

Относительный путь — это путь, который указывается относительно текущего рабочего каталога.

Команда `cd` (change directory) для перехода в директорию (папку)

Перейдите в папку YouTube по *абсолютному* пути:

`cd /D/2024/Investmate/Capital/YouTube`

Командой `pwd` проверьте куда вы переместились

Чтобы перейти на один уровень вверх

(в родительскую папку):

Введите команду `cd ..`

Чтобы перейти на два уровня вверх: `cd ../..`

Чтобы перейти на три уровня вверх: `cd ../../..` и т.д.

Командой `pwd` проверьте свое местонахождение

(в моем случае я дошла до корневой папки)



```
user@HOME-PC MINGW64 /d/2024/Investmate (main)
$ cd /D/2024/Investmate/Capital/YouTube

user@HOME-PC MINGW64 /D/2024/Investmate/Capital/YouTube (main)
$ pwd
/D/2024/Investmate/Capital/YouTube

user@HOME-PC MINGW64 /D/2024/Investmate/Capital/YouTube (main)
$ cd ..

user@HOME-PC MINGW64 /D/2024/Investmate/Capital (main)
$ cd ../..

user@HOME-PC MINGW64 /D/2024
$ cd ../../../../

user@HOME-PC MINGW64 /
$ pwd
/

user@HOME-PC MINGW64 /
```

Команды **cd**, **cd ~**, **cd /**, **cd -**

Вернитесь в папку Investmate, используя абсолютный путь:

cd /D/2024/Investmate

Введите команду **ls -l**

Перейдите в папку Learning по *относительному* пути:

cd Learning

Введите команду **pwd**

Перейдите в предыдущую директорию:

Введите команду **cd -**

Для информации :

Перейти в домашнюю директорию пользователя: **cd** или **cd ~**

Перейти в корневую директорию: **cd /**

Команда cp (copy) для копирования файлов и каталогов

1. Команда **cp file1.txt file2.txt** копирует файл file1.txt в новый созданный файл с именем file2.txt в текущей директории. Если file2.txt существует, то содержимое file2.txt будет перезаписано содержимым из file1.txt.
Введите в терминале **cp closeposition.json closeposition1.json**, введите **ls**
2. Команда **cp file1.txt /path/to/destination/** копирует файл file1.txt в каталог /path/to/destination/
Введите **cp closeposition.json /D/2024/Investmate/Trade/**
Введите **cd Trade, ls -la, cd .., pwd**
3. Команда **cp file1.txt file2.txt file3.txt file4.txt /path/to/destination/** копирует файлы file1.txt, file2.txt, file3.txt, file4.txt в каталог /path/to/destination/
Введите **cp lesson.txt quiz.txt trendpredictor.json tradingtip.json /D/2024/Investmate/Learning**
Введите **cd Learning, ls -la, cd .., pwd**
4. Команда **cp /path/to/file1.txt /path/to/destination/** копирует файл из указанного исходного пути в целевой каталог.
Введите в терминале **cp /D/2024/Investmate/script.sh /D/2024/Investmate/Settings.**
Введите **cd Settings, ls -la, cd .., pwd**
5. Команда **cp /path/to/file1.txt /path/to/file2.txt /path/to/destination** копирует файлы из разных директорий в целевую папку. Повторите команды, как указано ниже:

```
user@HOME-PC MINGW64 /D/2024/Investmate/Settings (main)
$ cp /D/2024/Investmate/Learning/lesson.txt /D/2024/Investmate/Trade/closeposition.json /D/2024/Investmate/Settings

user@HOME-PC MINGW64 /D/2024/Investmate/Settings (main)
$ ls -la
total 5
drwxr-xr-x 1 user 197121 0 Mar 18 21:33 ./
drwxr-xr-x 1 user 197121 0 Mar 18 15:10 ../
-rw-r--r-- 1 user 197121 0 Mar 18 21:40 closeposition.json
-rw-r--r-- 1 user 197121 7 Mar 18 21:40 lesson.txt
-rw-r--r-- 1 user 197121 0 Mar 18 20:47 reminder.txt
-rw-r--r-- 1 user 197121 0 Mar 18 20:46 script.sh
```

Команда `cp` (`copy`) для копирования файлов и каталогов

6. Команда **`cp /path/to/file1.txt /path/to/destination/new_file1.txt`** копирует файл из указанного исходного пути в целевой каталог с новым именем. Повторите команды, как указано ниже:

```
user@HOME-PC MINGW64 /d/2024/Investmate (main)
$ cp /d/2024/Investmate/mygoal.json /d/2024/Investmate/Trade/new_my_goal.json

user@HOME-PC MINGW64 /d/2024/Investmate (main)
$ ls
Capital/          favourites.txt    reminder.txt
Learning/         language.md      reset.txt
Settings/         lesson.txt       script.sh
Trade/            mygoal.json      tradinginstruments.json
closeposition.json openposition.txt  tradingtip.json
closeposition1.json quiz.txt          trendpredictor.json

user@HOME-PC MINGW64 /d/2024/Investmate (main)
$ cd Trade

user@HOME-PC MINGW64 /d/2024/Investmate/Trade (main)
$ ls
closeposition.json  new_my_goal.json

user@HOME-PC MINGW64 /d/2024/Investmate/Trade (main)
$
```

7. Команда **`cp -i file1.txt /path/to/destination/`** копирует файл, но перед заменой файла в целевом каталоге, запросит подтверждение у или n.

```
user@HOME-PC MINGW64 /D/2024/Investmate/Capital/YouTube/Learning (main)
$ cp -i lesson.txt /D/2024/Investmate
cp: overwrite '/D/2024/Investmate/lesson.txt'?
```


Команда `cp` (`copy`) для копирования файлов и каталогов

8. Команда **`cp -r /path/to/source/. /path/to/destination/`** копирует все файлы, включая скрытые (начинающиеся с точки), из исходного каталога в целевой каталог.

9. Команда **`cp *.txt /path/to/destination/`** копирует все файлы с расширением `.txt` в указанный каталог.

Вернитесь в терминале в папку `Investmate`, используя абсолютный путь или команду **`cd ..`**

Повторите команды, как указано ниже:

```
user@HOME-PC MINGW64 /D/2024/Investmate (main)
$ cp *.txt /D/2024/Investmate/Capital/YouTube/Learning

user@HOME-PC MINGW64 /D/2024/Investmate (main)
$ cd /D/2024/Investmate/Capital/YouTube/Learning

user@HOME-PC MINGW64 /D/2024/Investmate/Capital/YouTube/Learning (main)
$ ls -l
total 1
-rw-r--r-- 1 user 197121 0 Mar 18 16:32 favourites.txt
-rw-r--r-- 1 user 197121 7 Mar 18 16:32 lesson.txt
-rw-r--r-- 1 user 197121 0 Mar 18 16:32 openposition.txt
-rw-r--r-- 1 user 197121 0 Mar 18 16:32 quiz.txt
-rw-r--r-- 1 user 197121 0 Mar 18 16:32 reminder.txt
-rw-r--r-- 1 user 197121 0 Mar 18 16:32 reset.txt
-rw-r--r-- 1 user 197121 0 Mar 18 15:56 tradingtip.json
-rw-r--r-- 1 user 197121 0 Mar 18 15:56 trendpredictor.json
```

Переместиться в папку `Learning` можно и по относительному пути:

```
user@HOME-PC MINGW64 /d/2024/Investmate (main)
$ cd ./Learning
```

Команда **cp** (**copy**) для копирования файлов и каталогов

10. Команда **cp -p file1.txt /path/to/destination/** копирует файл с сохранением его прав доступа (права на чтение/запись) и меток времени.

11. Команда **cp -v file1.txt /path/to/destination/** копирует файл и выводит информацию о процессе копирования (опция **-v** для подробного вывода).

12. Команда **cp -a /path/to/source /path/to/destination** копирует каталог и его содержимое, сохраняя символические ссылки, если они есть (опция **-a**).

13. Команда **cp -f file1.txt /path/to/destination/** принудительно копирует файл, заменяя файл в целевом каталоге без запросов на подтверждение.

14. Команда **cp --preserve=timestamps file1.txt /path/to/destination/** копирует файл с сохранением временных меток (время последнего доступа и изменения).

15. Команда **cp -u /path/to/file1.txt /path/to/destination/** копирует файл только **если он новее** (то есть если его дата модификации более поздняя), либо если файл в целевой папке **отсутствует** (**-u** параметр **update**).

16. Команда **cp -r /path/to/source/ /path/to/destination/** копирует каталог **source** и все его содержимое в каталог **destination** (опция **-r** рекурсивно копирует каталоги).

Введите **cp -r /D/2024/Investmate/Learning /D/2024/Investmate/Capital/YouTube**

Введите **pwd, ls -la, cd Capital, ls, cd YouTube, ls, cd Learning, ls -la**

Команда mv для перемещение файлов и каталогов

Если файл нужно переместить, а не копировать, то вместо команды **cp** используйте команду **mv (move)**

Для перемещения файла можно также использовать относительные и абсолютные пути.

Чтобы переместить файл в папку в рамках текущей директории используйте команду **mv file_name target_directory/**

Вернитесь в терминале в папку Investmate, используя абсолютный путь или команду **cd ..**

Введите **mv closeposition.json openposition.txt tradinginstruments.json reset.txt /D/2024/Investmate/Trade**

```
Capital/      favourites.txt  reminder.txt
Learning/     language.md    reset.txt
Settings/     lesson.txt     script.sh
Trade/        mygoal.json    tradinginstruments.json
closeposition.json  openposition.txt  tradingtip.json
closeposition1.json  quiz.txt         trendpredictor.json

user@HOME-PC MINGW64 /d/2024/Investmate (main)
$ mv closeposition.json openposition.txt tradinginstruments.json reset.txt /D/2024/Investmate/Trade

user@HOME-PC MINGW64 /d/2024/Investmate (main)
$ ls
Capital/  Trade/      language.md  quiz.txt    tradingtip.json
Learning/ closeposition1.json  lesson.txt   reminder.txt trendpredictor.json
Settings/ favourites.txt  mygoal.json  script.sh

user@HOME-PC MINGW64 /d/2024/Investmate (main)
$ cd Trade

user@HOME-PC MINGW64 /d/2024/Investmate/Trade (main)
$ ls
closeposition.json  openposition.txt  tradinginstruments.json
new_my_goal.json   reset.txt
```

- Переместите файлы: mygoal.json, favourites.txt, reminder.txt, language.md в папку Settings
- Удалите из папки Investmate файлы: closeposition1.json, quiz.txt, tradingtip.json, lesson.txt, trendpredictor.json
- В папке Capital удалите папку YouTube
- Переместите script.sh в папку Capital
- Переименуйте в папке Capital new_my_goal.json на webinars.json
- Удалите в папке Trade new_my_goal.json
- Удалите в папке Settings closeposition.json lesson.txt