### Для запуска Linux в командной строке Windows 10 bash

### 

### 

### **1.Основы работы с командной строкой**

Структура команды Linux:

команда [-опции] [аргументы]

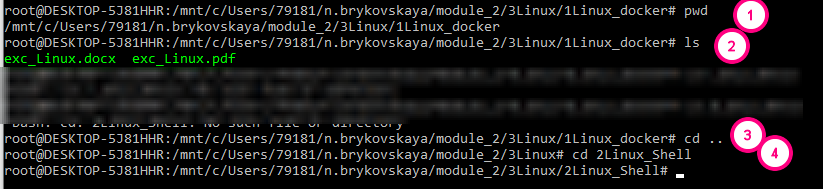
-команда: действие, которое нужно выполнить (например, ls, cd, mkdir).

--опции: модификаторы поведения команды (например, ls -l, mkdir -p).

-аргументы: объекты, над которыми выполняется команда (например, ls /home/user, cd Documents).

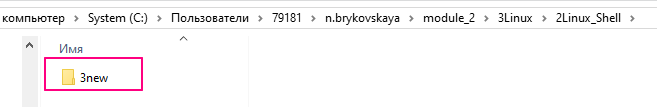
**Основные команды**

* pwd: отображает текущий рабочий каталог.
* ls: отображает содержимое каталога.
* cd: изменяет текущий рабочий каталог.

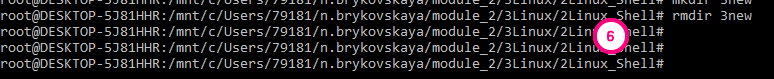


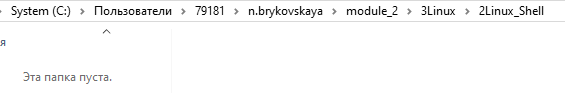
* mkdir: создает новый каталог.



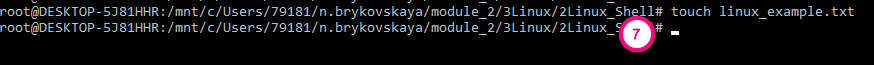


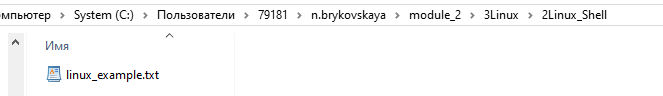
* rmdir: удаляет пустой каталог.





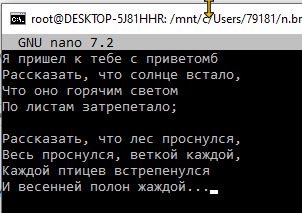
* touch: создает пустой файл.



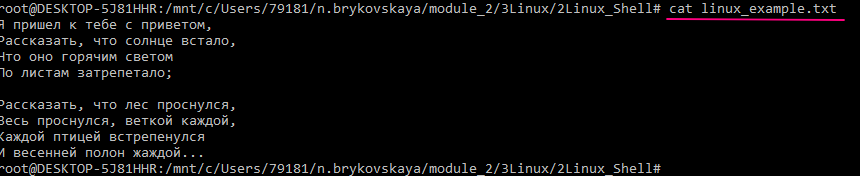


внесем изменения в текстовый файл

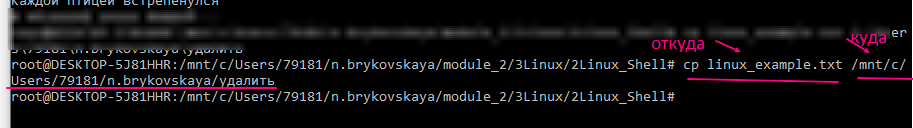


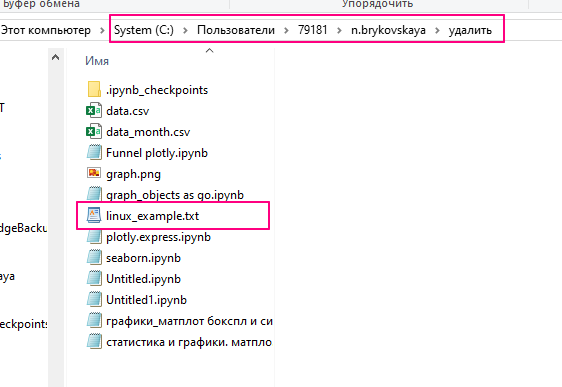
 еще раз исправим опечатки сохраним

* cat: отображает содержимое файла.

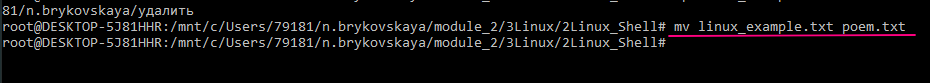


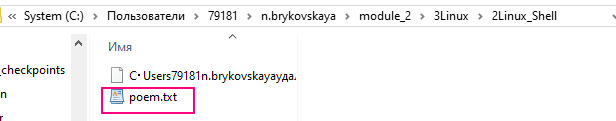
* cp: копирует файлы и каталоги.





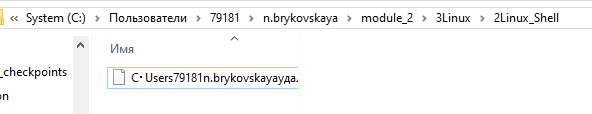
* mv: перемещает или переименовывает файлы и каталоги.





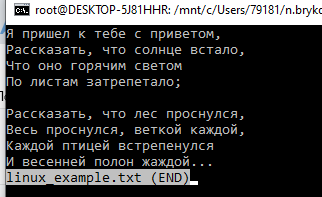
* rm: удаляет файлы и каталоги.



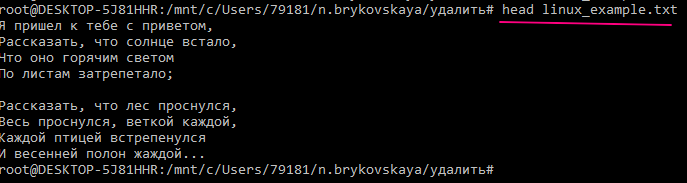


* less: просмотр содержимого файла с постраничной навигацией.

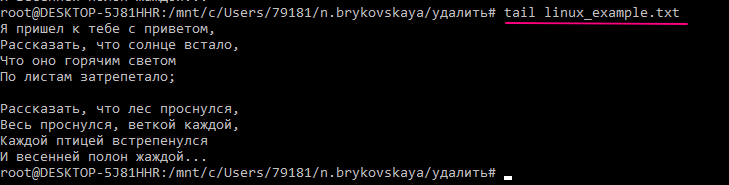


 **Чтобы выйти из команды less в Linux, нужно нажать клавишу q**. Это немедленно закроет просмотрщик less и вернёт к приглашению оболочки. q

* head: отображает первые строки файла.



* tail: отображает последние строки файла.

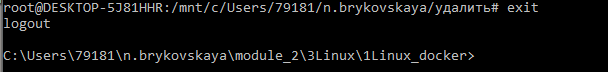


* grep: поиск строк, соответствующих заданному шаблону.

Стандартно используются базовые регулярные выражения (BRE), а для работы с расширенными регулярными выражениями (ERE) нужно явно прописывать настройки. Базовые выражения предоставляют определенный набор символов: **.**, **\***, **[]**, **[^]**, **^** и **$**.



выходим



### **Практика - Основные команды**

* Попробуйте выполнить каждую из перечисленных команд в терминале.
* Используйте флаги и аргументы, чтобы изменить поведение команд.
* Создайте несколько каталогов и файлов. Перемещайте и копируйте их.
* Просмотрите содержимое файлов с помощью cat, less, head и tail.
* Используйте grep для поиска информации в файлах.

все это проделано выше

### **2.Управление файлами и директориями**

* Абсолютный и относительный пути:

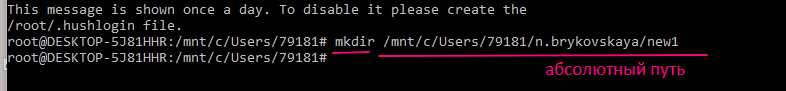
-Абсолютный путь: путь к файлу или каталогу, начиная с корневого каталога (/).

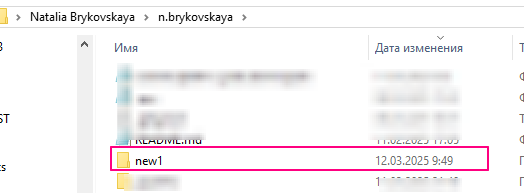
-Относительный путь: путь к файлу или каталогу, начиная с текущего рабочего каталога.

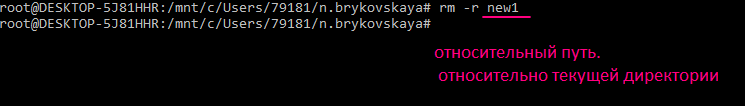
* Права доступа к файлам: Linux использует систему прав доступа, чтобы определить, кто может читать, записывать и выполнять файлы. Команда **chmod** (сокращение от change mode) используется для **изменения прав доступа к файлам и каталогам** в Unix-подобных операционных системах

### **Практика - Управление файлами и директориями**

* Попрактикуйтесь в использовании абсолютных и относительных путей при работе с командами.

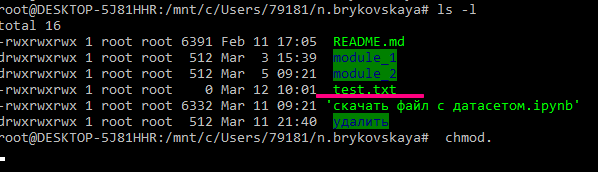






Подняться на уровень выше: cd ..

Если нужно поднять на два уровня выше, то cd ../..

* Изучите права доступа к файлам с помощью команды ls выводит список файлов и каталогов с подробной информацией о пользователях, правах доступа, размерах, владельцах и т. д. -l вывести длинный список с подробной информацией.

В Linux права доступа к файлам и папкам делятся на три категории:

**Владелец (Owner)** — пользователь, который создал файл или папку.

**Группа (Group) —** группа пользователей, которые могут иметь доступ к файлу или папке.

**Все остальные (Others**) — все остальные пользователи системы.

Каждый из этих уровней имеет три типа прав:

**Чтение (Read, r)** — разрешает просматривать содержимое файла или списка файлов в директории.

**Запись (Write, w)** — разрешает изменять содержимое файла или добавлять/удалять файлы в директории.

**Исполнение (Execute, x**) — разрешает запускать файл как программу или входить в директорию.

* Измените права доступа к файлам с помощью команды chmod.



устанавливает права доступа к файлу test.txt чтение, запись и выполнение для владельца, группа — только читать, остальные — только читать.

ды

### **3.Работа с текстовыми файлами**

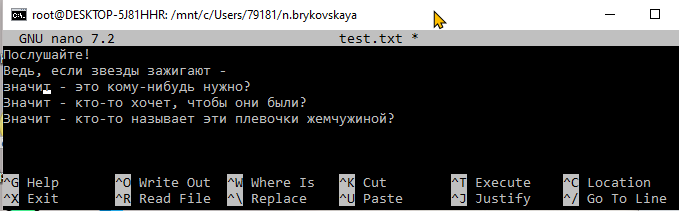
* Редакторы текста в терминале: nano: простой и удобный редактор для начинающих.
* vim: мощный и расширяемый редактор для опытных пользователей.

### **Практика - Работа с текстовыми файлами**

* Создайте текстовый файл с помощью команды touch.



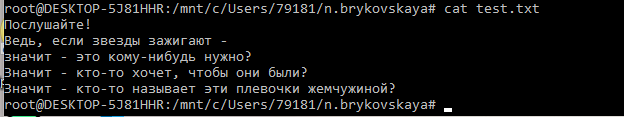
* Откройте файл в редакторе nano и напишите несколько строк текста.



* Сохраните файл и закройте редактор.



* Просмотрите содержимое файла с помощью команды cat.



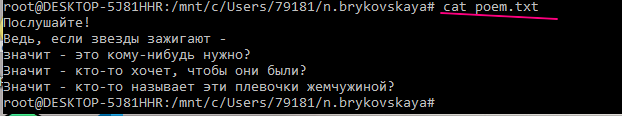
### **4.Расширенные возможности командной строки**

* Перенаправление ввода/вывода:

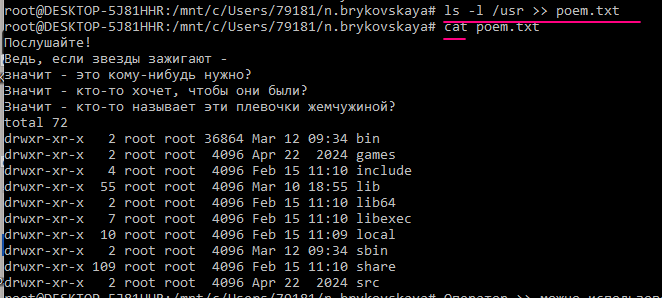
> перенаправление вывода команды в файл.



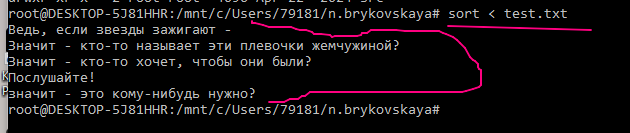
проверим



>> добавление вывода команды в конец файла



< перенаправление ввода команды из файла.



| перенаправление вывода одной команды на вход другой (конвейер).

* Использование символов-шаблонов:

\* заменяет любую последовательность символов.



? заменяет один любой символ.



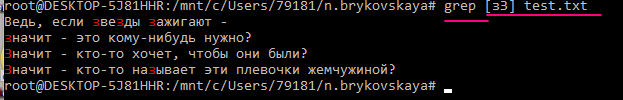
[] заменяет один символ из указанного диапазона.

Один из примеров — команда, которая выводит на экран все файлы и подкаталоги текущей директории, названия которых начинаются с буквы p, q, r или s и имеют произвольное количество символов: ls [p-s]\*.









### **Практика - Расширенные возможности командной строки**

* Попрактикуйтесь в перенаправлении ввода/вывода с помощью команд ls, cat, grep.
* Используйте символы-шаблоны для поиска файлов и каталогов.

все это проделано выше

### **5.Создание и выполнение скриптов Bash**

* Скрипт Bash: текстовый файл, содержащий последовательность команд Linux.
* Shebang: первая строка скрипта, указывающая интерпретатор (#!/bin/bash).
* Запуск скрипта: сделать скрипт исполняемым (chmod +x script.sh) и запустить его (./script.sh).

### **Практика - Создание и выполнение скриптов Bash**

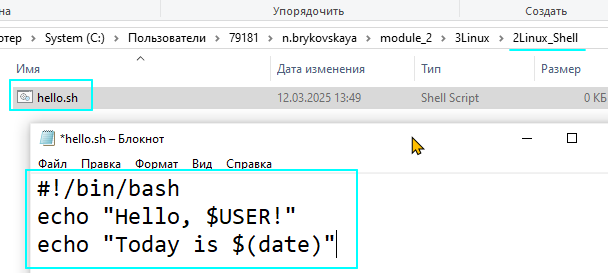
* Создайте простой скрипт Bash, который выводит текст на экран.

пример оператора шебанг #!/bin/bash

путь к оболочке bash (который может отличаться от указанного выше) с помощью команды:



создадим файл в блокноте

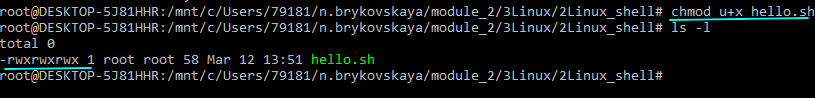


Строка #1: Шебанг (#!/bin/bash) указывает на путь к оболочке bash.

* Сделайте скрипт исполняемым и запустите его.

Чтобы сделать скрипт исполняемым, назначьте права на выполнение своему пользователю с помощью команды: chmod u+x hello.sh Где,

* chmod изменяет владельца файла для текущего пользователя u:.
* +x добавляет права на выполнение текущему пользователю. Это означает, что пользователь, являющийся владельцем, теперь может запустить сценарий

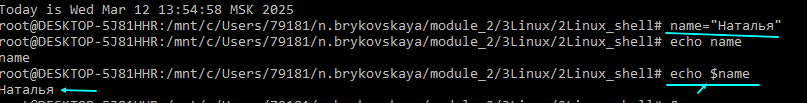


hello.sh — это файл, который мы хотим запустить.

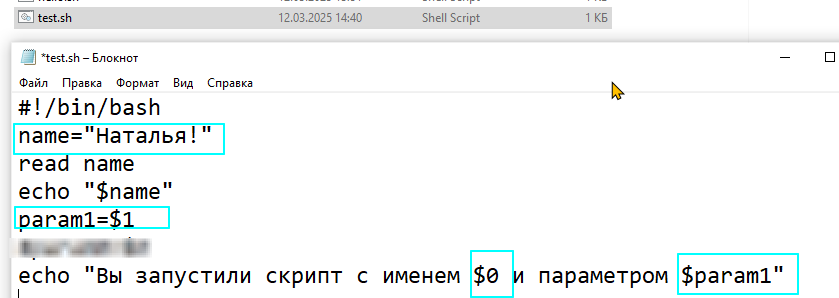


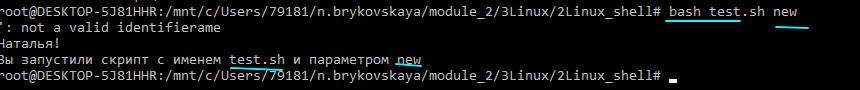
* Добавьте в скрипт переменные, условия и циклы.

Для создания переменной в Bash необходимо присвоить ей значение, используя знак равенства без пробелов (=). Например : name="Наталья"



добавим в скрипт test.sh переменные и присвоим им параметры скрипта





### **6.Управление процессами**

* Процесс: запущенная программа.
* Команды для управления процессами:

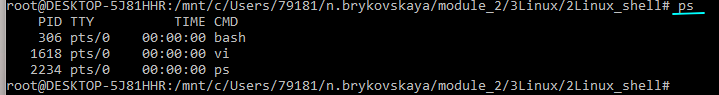
ps отображает список запущенных процессов.

top отображает динамический список процессов.

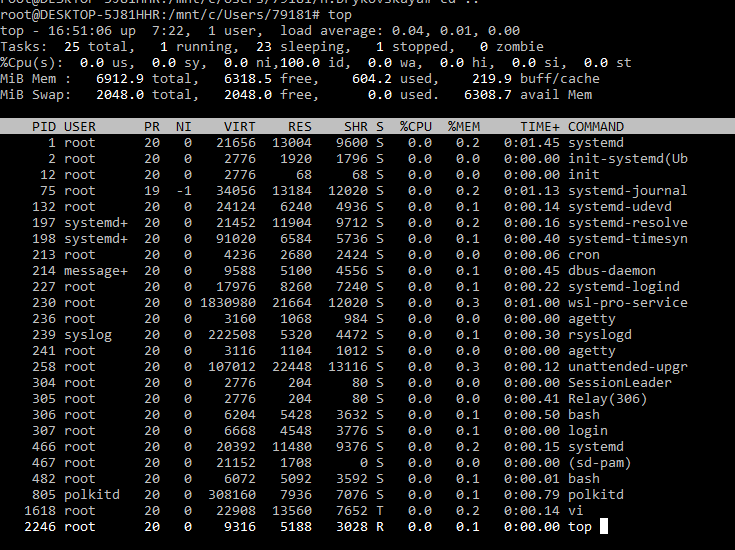
kill отправляет сигнал процессу (например, завершение работы).

### **Практика - Управление процессами**

* Используйте команду ps для просмотра списка запущенных процессов.



* Используйте команду top для наблюдения за ресурсоемкими процессами.



* Завершите процесс с помощью команды kill.



### **7.Подключение к удаленному серверу по SSH**

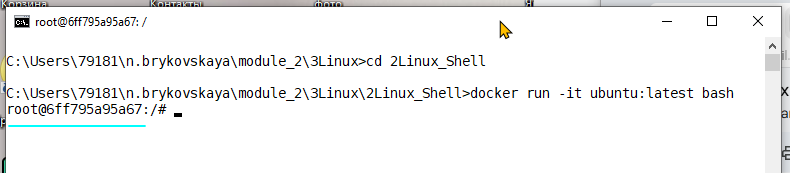
* SSH (Secure Shell): протокол для безопасного удаленного управления компьютером.
* SSH-ключ: пара файлов (открытый и закрытый ключ), используемая для аутентификации.

### **Практика - Подключение к удаленному серверу по SSH**

* Сгенерируйте пару SSH-ключей на вашем компьютере.

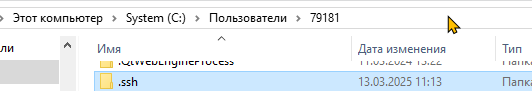
в командной строке на личном ПК сформирована пара SSH-ключей с помощью команды ssh-keygen

================



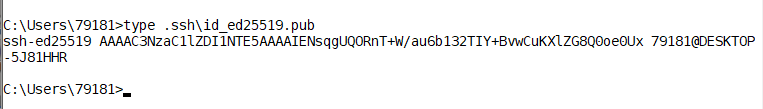
====================





файлы сохранены тут C:\Users\79181\.ssh\id\_ed25519.pub

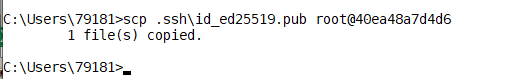
выведем на экран



* Скопируйте открытый ключ в контейнер (удаленный сервер).

для этого

Скопируйте открытый ключ на другой компьютер с помощью команды scp. Например, из домашнего каталога запустите команду: scp .ssh/id\_rsa.pub имя пользователя@other\_machine



Введите пароль для удалённого компьютера, файл будет передан.

Войдите в систему на другом компьютере с именем пользователя и паролем, чтобы поместить файл в нужное место.

Убедитесь, что домашний каталог на другом компьютере содержит каталог с именем .ssh с теми же правами доступа, что и на исходном компьютере.

$ mkdir .ssh

$ chmod 700 .ssh

В этом каталоге должен быть файл с именем authorized\_keys, доступный только пользователю.

Добавьте открытый ключ, который передали, в этот файл: cat id\_rsa.pub >> .ssh/authorized\_keys.

* Подключитесь к контейнеру (удаленному серверу) по SSH, используя ваш закрытый ключ.