**Занятие 2. Основы командной строки в Linux.**

### **Краткая история открытых операционных систем**

В конце 1960-х и начале 1970-х годов в исследовательском подразделении американской телекоммуникационной корпорации AT&T была создана операционная система UNIX. В то время уже существовало более сотни различных операционных систем, и около 15 из них активно использовались.

UNIX, разработанная Кеном Томпсоном при участии нескольких коллег, отличалась от предшественников благодаря следующим ключевым характеристикам:

* **Компонентная архитектура**: принцип «одна программа — одна функция» и мощные средства для связывания программ для решения сложных задач.
* **Минимизация ядра**: небольшое ядро и минимальное количество системных вызовов, что делает систему гибкой.
* **Независимость от аппаратной архитектуры**: UNIX была реализована на языке программирования C, что позволило легко переносить её на разные платформы.

UNIX быстро завоевала популярность, сначала в университетах, а затем и в отрасли, где она стала универсальной системой для разных аппаратных платформ.

#### **Основные ветви UNIX:**

1. **BSD (Berkeley Software Distribution)** — вариант UNIX, разработанный в Калифорнийском университете в Беркли, который внёс значительные улучшения и дополнения к системе.
2. **GNU** — проект, основанный Ричардом Столлманом, целью которого было создать свободную альтернативу UNIX. Основной результат проекта — утилиты и средства, используемые в операционных системах BSD, UNIX и Linux.
3. **Linux** — операционная система, основанная на ядре, разработанном Линусом Торвальдсом в начале 1990-х годов. Она стала основой для множества дистрибутивов, таких как Debian, Red Hat и другие.

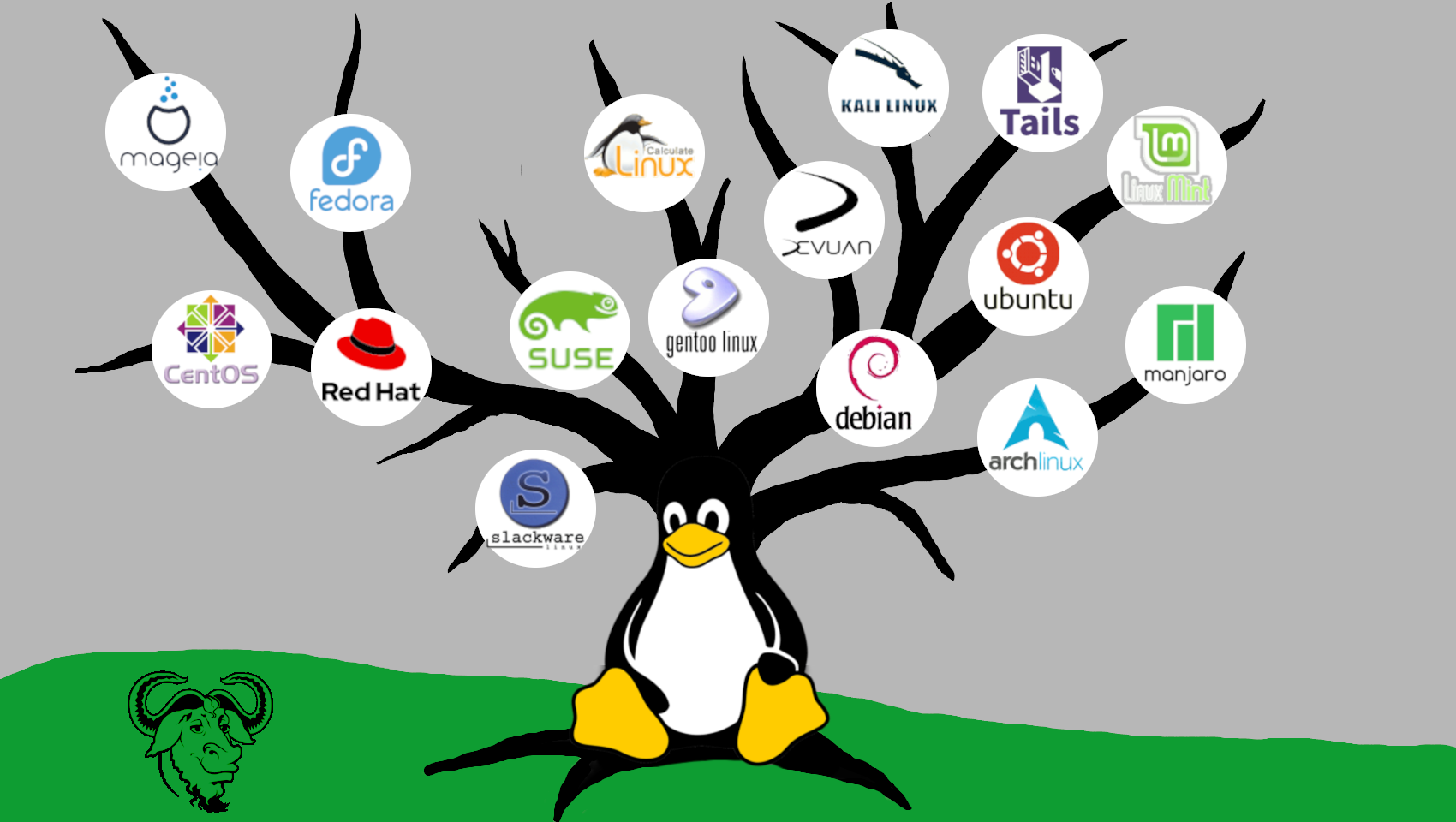
### **Современные семейства открытых операционных систем:**

1. **UNIX-подобные системы от AT&T** (AIX, Solaris).
2. **BSD-системы** (FreeBSD, OpenBSD, NetBSD, Darwin — основа Mac OS X).
3. **GNU/Linux** — операционные системы на основе ядра Linux и компонентов проекта GNU (Debian, RedHat, Mandriva и другие).

Linux-системы обычно распространяются в виде **дистрибутивов**, которые включают ядро, библиотеки и утилиты проекта GNU, а также программы установки и дополнительные утилиты.

Ссылки:

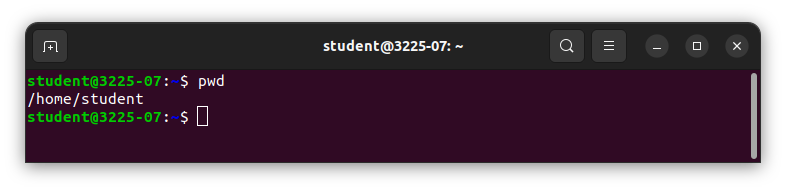
* [История дистрибутивов Linux (диаграмма)](https://upload.wikimedia.org/wikipedia/commons/1/1b/Linux_Distribution_Timeline.svg)



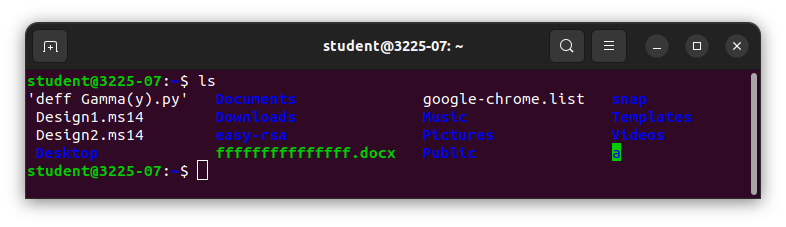
##### **Основы командной строки**

##### **Работа с директориями**

**pwd**— полный путь до папки, в которой находитесь сейчас:

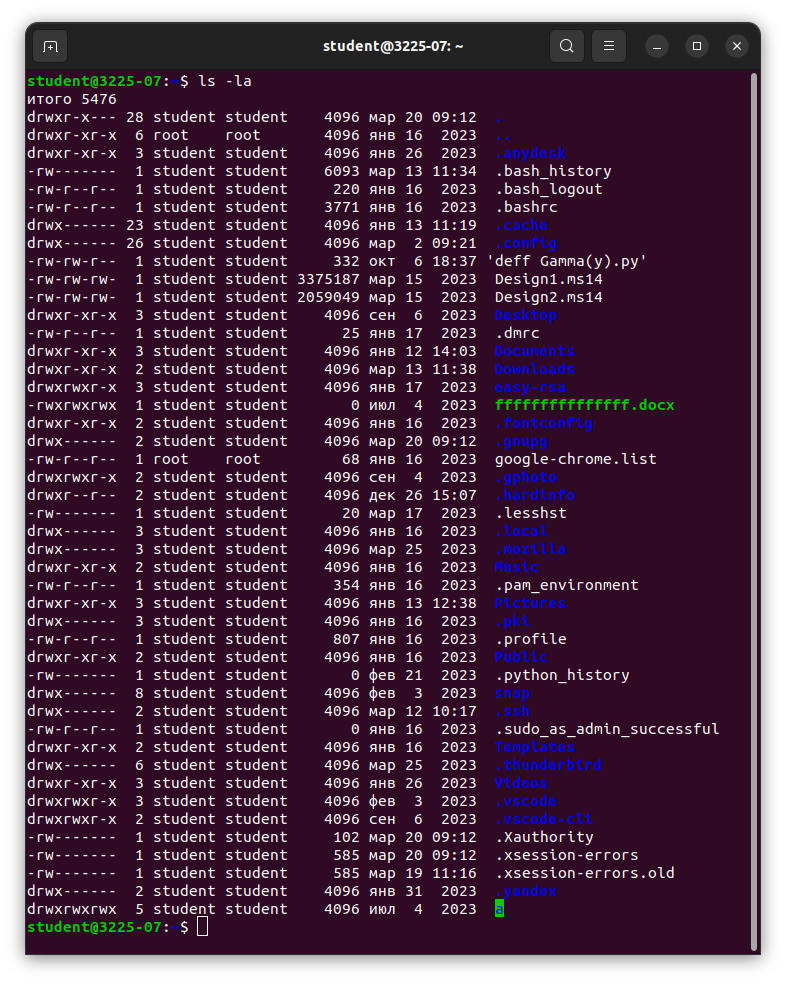


**ls**— просмотр содержимого открытой папки:

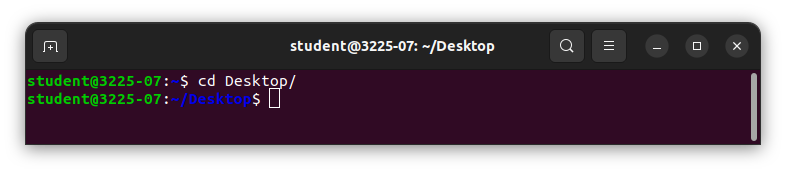


У команд есть ключи — например, можно написать команду ls с ключом -a и увидеть список всех файлов, в том числе скрытых. Как правило, у всех команд терминала есть такие дополнительные ключи. Их может быть довольно много — например, как в списке ключей для команды ls.

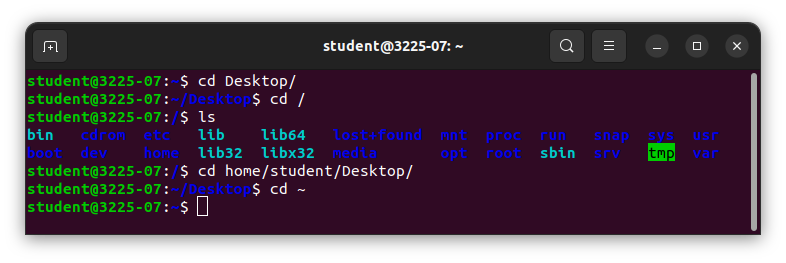
**ls -la**выводит дополнительную информацию (о правах, владельце, дате редактирования и размере).



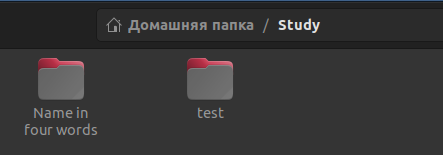
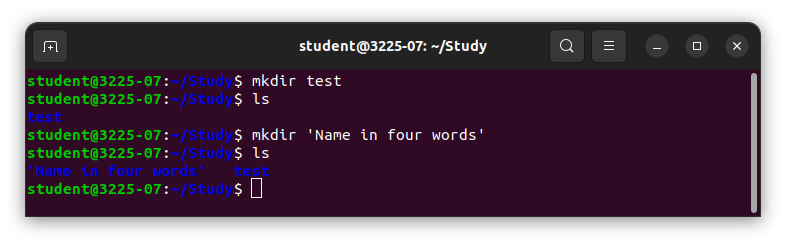
**cd**название папки прокладывает навигацию к любой папке. Это происходит примерно по той же схеме, что в проводнике или Finder.



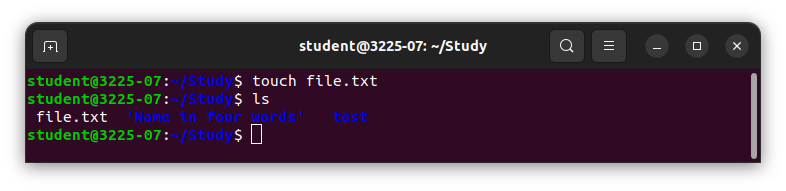
|  |  |
| --- | --- |
| Символы | На что направляет |
| **.** | Текущая директория |
| **..** | Директория, находящаяся выше |
| **/** | Корневая директория |
| **~** | Домашняя директория |



**mkdir**название директории — создаёт папку. Если у директории в названии присутствует несколько символов, название пишется в кавычках:



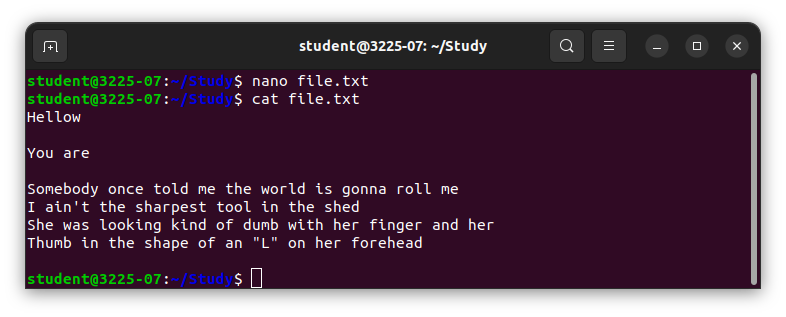
**touch название файла** — создаёт файл, а также указывает расширение:



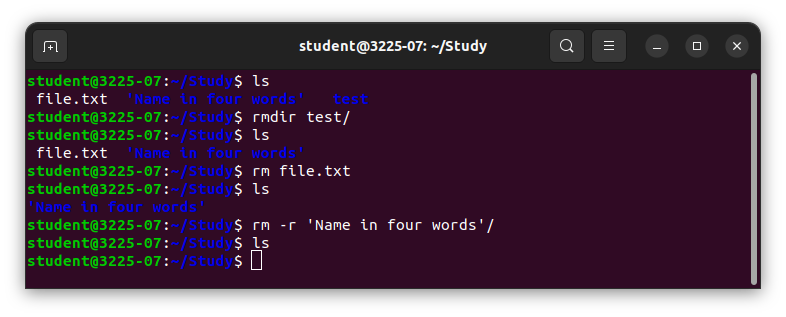
**nano название**— текстовый редактор файла:



**cat название**— построчно выводит содержимое файла:

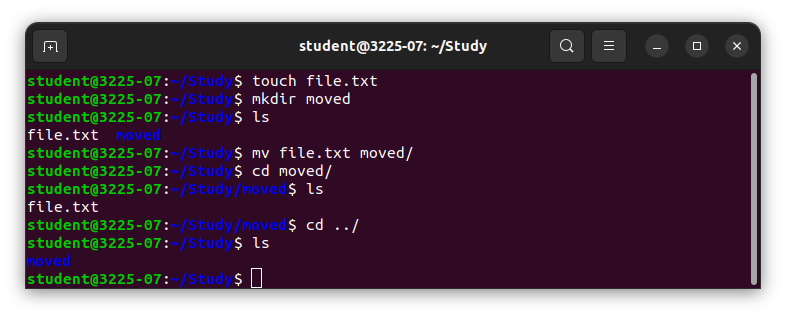


**rmdir название** — удаляет директорию. Команда **rm название** удаляет файл. Также можно удалять директорию при помощи **rm -r название** :



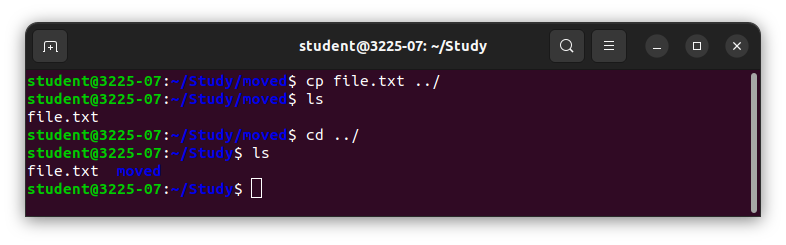
**mv** - перемещение файлов

Команда mv перемещает файл в новое место. Она также может использоваться для переименования файлов. Чтобы переместить файл в другую папку нужно указать путь к ней.



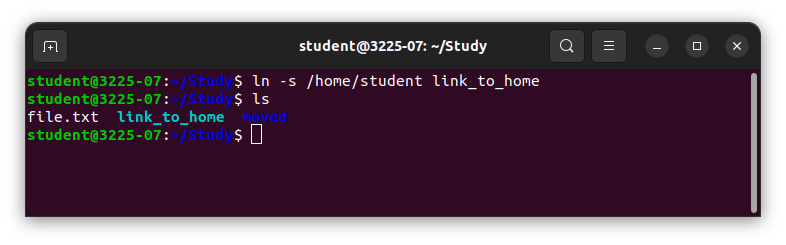
**cp** - копирование файлов

Эта cp и mv - это похожие команды linux для работы с файлами. Они работают аналогичным образом, только исходный файл остается на своем месте. Вы также можете рекурсивно скопировать всю папку с помощью команды cp -r. Эта команда скопирует всю папку вместе со всеми файлами и вложенными папками в новое место.



**ln** - создание ссылок

Утилита ln позволяет создавать жесткие и символические ссылки на файлы или папки. Для создания символической ссылки используется опция -s. Например, мы создадим ссылку на папку с загрузками:



|  |  |
| --- | --- |
| **Команда** | **Что делает** |
| **pwd** | Выводит текущую директорию |
| **ls** | Выводит список содержимого |
| **cd** | Переход в другую директорию |
| **mkdir** | Создание директории |
| **touch** | Создание файла |
| **ln** | Создание ссылки |
| **nano** | Редактор файлов |
| **cat** | Выводит содержимое файла построчно |
| **mv** | Перенос / переименовывание файла |
| **cp** | Копирование файла, директории |
| **rm** | Удаление файла / директории |
| **rmdir** | Удаление директории |
| **man** | Описание команды |

##### **Перенаправление ввода и вывода**

Ввод и вывод распределяется между тремя стандартными потоками:

stdin — стандартный ввод (клавиатура),

stdout — стандартный вывод (экран),

stderr — стандартная ошибка (вывод ошибок на экран).

< file — использовать файл как источник данных для стандартного потока ввода.

> file — направить стандартный поток вывода в файл (перезапись)

2> file — направить стандартный поток ошибок в файл (перезапись)

>>file — направить стандартный поток вывода в файл (добавление)

2>>file — направить стандартный поток ошибок в файл. (добавление)

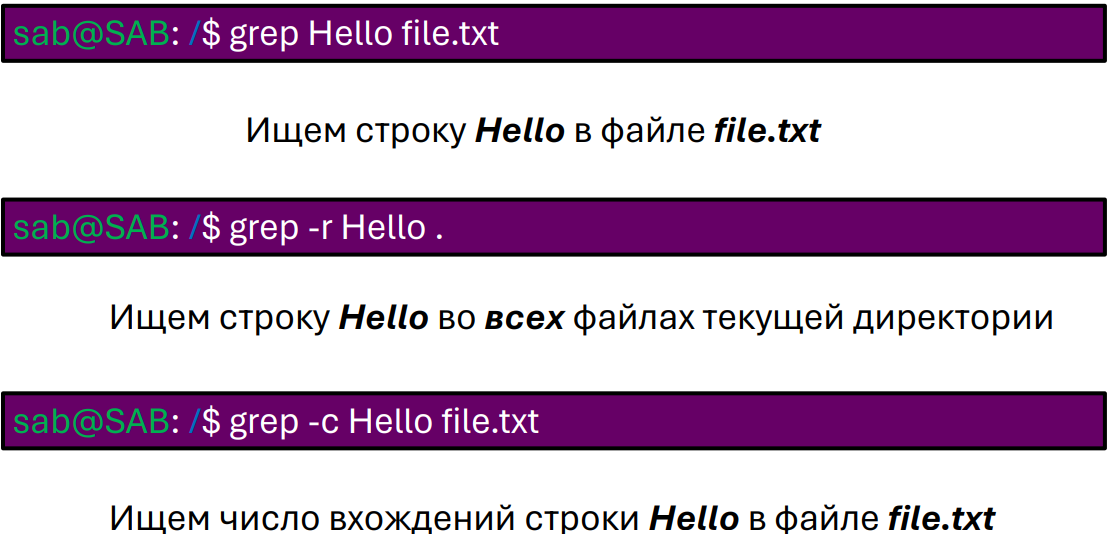
&>file или >&file — направить с.п. вывода и с.п. ошибок в файл.

##### **Команды поиска**

**Grep** (global regular expression printer) – утилита командной строки, позволяющая производить поиск строки в файле.

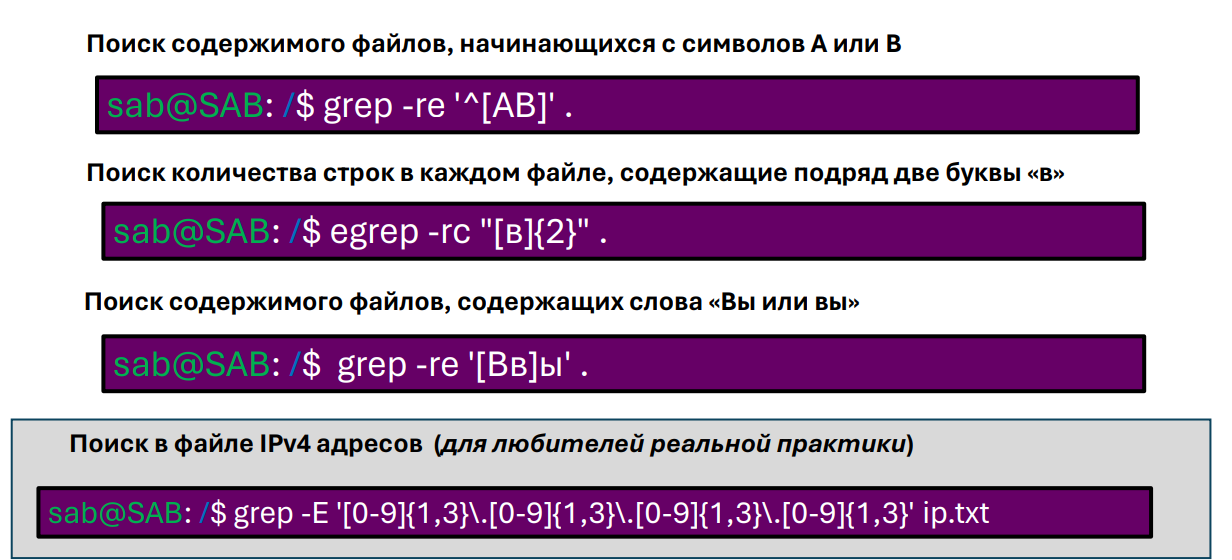


|  |  |
| --- | --- |
| **Ключ** | **Описание** |
| **-c** | Выдает только количество строк, содержащих выражение. |
| **-h** | Скрывает вывод названия файла, в котором было обнаружено вхождение. Используется при поиске по нескольким файлам. |
| **-i** | Игнорирует регистр символов при поиске. |
| **-l** | Выдает только имена файлов, содержащих сопоставившиеся строки |
| **-n** | Выдает перед каждой строкой ее номер в файле (строки нумеруются с 1). |
| **-s** | Скрывает выдачу сообщений о не существующих или недоступных для чтения файлах. |
| **-v** | Выдает все строки, за исключением содержащих выражение. |
| **-E** | Поиск с использованием регулярных выражений |
| **-o** | Вывод только обнаруженных символов |
| **-r** | Рекурсивный поиск |

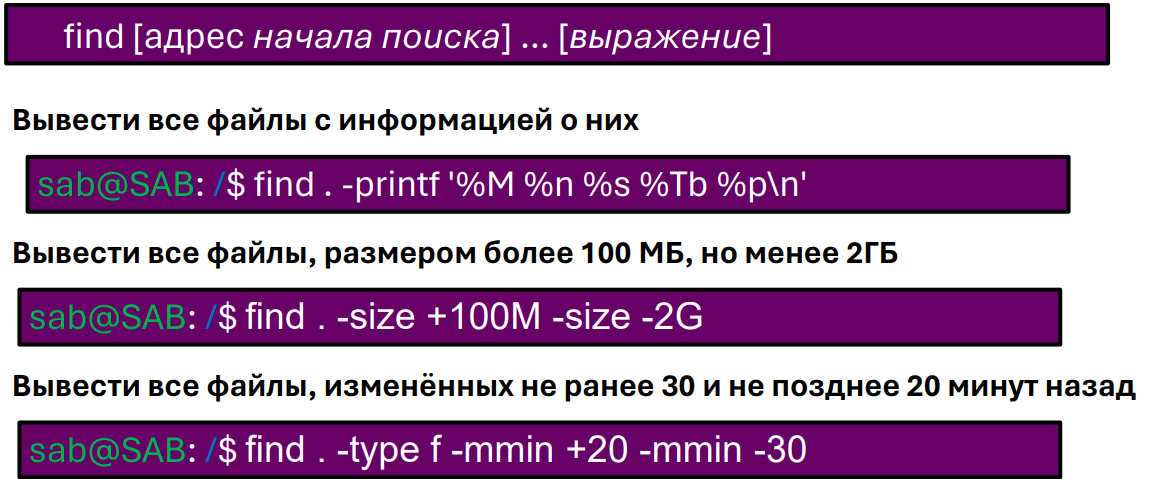


##### **Регулярные выражения**

|  |  |
| --- | --- |
| **Метасимвол** | **Описание** |
| **\** | начало буквенного спецсимвола |
| **^** | указывает на начало строки |
| **$** | указывает на конец строки |
| **\*** | указывает, что предыдущий символ может повторяться 0 или больше раз |
| **+** | указывает, что предыдущий символ должен повторится больше один или больше раз |
| **?** | предыдущий символ может встречаться ноль или один раз |
| **{n}** | указывает сколько раз (n) нужно повторить предыдущий символ |
| **{N,n}** | предыдущий символ может повторяться от N до n раз |
| **.** | любой символ кроме перевода строки |
| **[az]** | любой символ, указанный в скобках |
| **x|y** | символ x или символ y |
| **[^az]** | любой символ, кроме тех, что указаны в скобках |
| **[a-z]** | любой символ из указанного диапазона |
| **[^a-z]** | любой символ, которого нет в диапазоне |
| **[[:alpha:]]** | является алфавитным символом |
| **[[:digit:]]** | является числом |



**find** – утилита, с помощью которой возможно найти файл по его имени



**cut** – утилита, с помощью которой возможно вырезать символы из каждой переданной ей строки

**tr** (translate) – утилита командной строки, позволяющая посимвольно обрабатывать текст

**wc** (word count) – утилита командной строки, выводящая число переводов строк, слов и байт для каждого указанного файла и итоговую строку, если было задано несколько файлов.

**uniq** – утилита командной строки, предназначена для поиска одинаковых строк в массивах текста.

**awk** – скриптовый язык построчного разбора и обработки входного потока

##### **Полезные команды для повседневки**

**hexdump** — показывает шестнадцатеричное представление данных, поступающих на стандартный поток ввода.

**cat** — считывает данные со стандартного потока ввода и передает их на стандартный поток вывода.

**sudo** - запуск программы от имени других пользователей, а также от имени суперпользователя.

**bc** – калькулятор

**python3** – среда разработки Python

**sort** – сортировка переданных значений (-r – обратный порядок)

##### **Прикольные команды**

https://losst.pro/prikolnye-komandy-linux?ysclid=lu027tik2g297486326

##### **Задания**

**Задание 1.**

1. Перейдите в директорию lab1.

2. Оставаясь в директории lab1, создайте в каталоге poems/English файл, содержащий текст вашего любимого стихотворения отечественного автора. Название файла должно соответствовать названию стихотворения. Внутри файла, перед текстом произведения укажите название и автора.

3. Перенесите созданный файл из директории English в Russian.

4. Создайте в директории English каталоги, содержащие названия веков и распределите по ним расположенные в ней стихотворения.

**Задание 2.**

1. Произведите поиск всех стихотворений, названия которых содержит только кириллицу

2. Найдите все файлы с расширением jpeg

3. Найдите все файлы, которые были изменены за последние 20 минут

4. Найдите все файлы, объемом больше 500 Кб

**Задание 3.**

1. Вычислите у скольких стихотворений вместо названия стоят “\*\*\*”

2. Выведите напротив каждого файла сообщение о том, содержит ли он восклицательный знак

3. Рассчитайте, сколько раз в тексте стихотворений встречается предлог «на»

4. Вычислите самое часто встречающееся слово в монологе Гамлета