МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем

и технологий

**Отчет**

**по лабораторной работе № 3**

по дисциплине: ”Системное программирование”

на тему: ***”*****Командный язык и скрипты Shell*”***

Выполнил**:** студент группы *10702121* Агеенко А. К.

Высоцкий М. Л.

Градусов А. Р.

Принял**:** пр. Давыденко Н. В.

Минск 2023

# Лабораторная работа №3.

**Цель работы:** Закрепить на практике основы работы со сценариями командной оболочки и перенаправление ввода/вывода.

## Задание 1

1. Написать скрипт, который записывает содержимое одного файла в другой и при этом оба файла существуют до операции. Дать два разных варианта выполнения.

2. Найти все файлы, начинающиеся на слово script и записать их имена в файл list.txt.

3. Написать скрипт, который проверяет, содержится ли скриптовый файл в директории, если да, то выполнить его.

4. Написать скрипт, подсчитывающий сумму от 1 до 10.

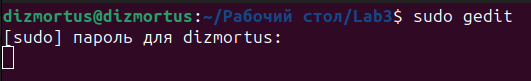
5. А какие командные интерпретаторы установлены в вашей системе?

Напишите скрипт поиска местонахождения командных интерпретаторов вашей системы. Также определите какой интерпретатор используется сейчас в вашей системе?

Решение

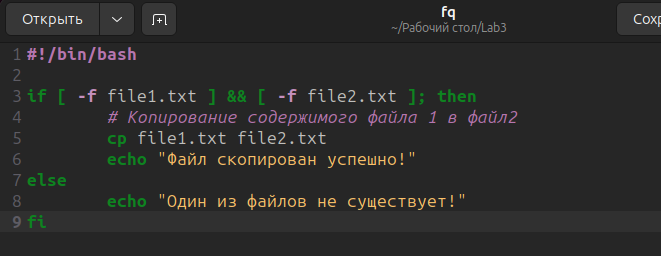
**1.** Написать скрипт, который записывает содержимое одного файла в другой и при этом оба файла существуют до операции. Дать два разных варианта выполнения.

1. Вызываем текстовый редактор

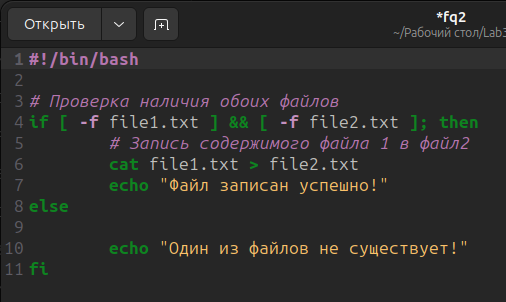


1. Пишем код сценария и сохраняем файл

1) Используя команду `cp` для копирования файла:



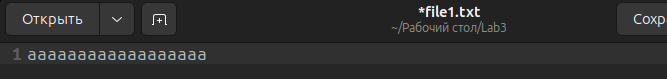
2) Используя оператор `cat` для чтения и записи файлов:



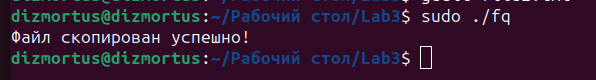
1. Делаем скрипт выполняемым файлом



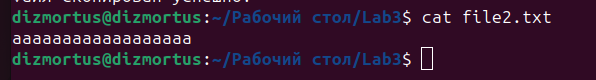
4. Создаем два текстовых файла и заносим текст в один из них.



5. Запускаем скрипт



6. Проверяем наш второй файл:

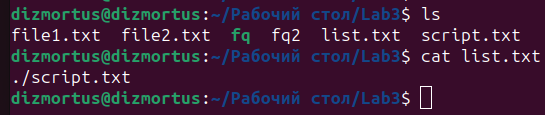


**2.** Найти все файлы, начинающиеся на слово script и записать их имена в файл list.txt.

1. Выполняем команду find

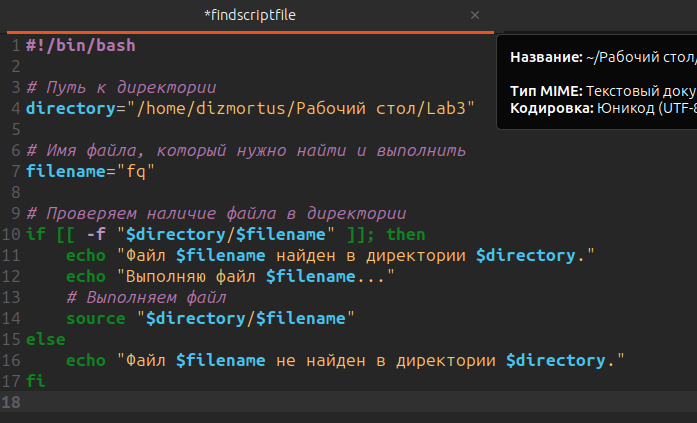


2. Проверяем результат

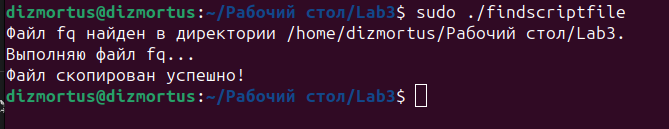


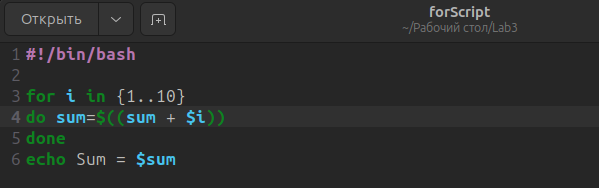
**3.** Написать скрипт, который проверяет, содержится ли скриптовый файл в директории, если да, то выполнить его.

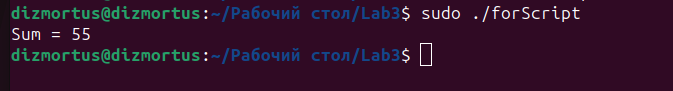
1. Пишем скрипт.



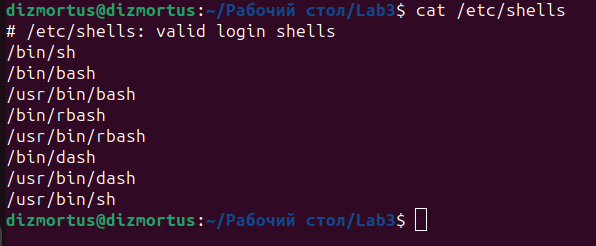
2. Запускаем и проверяем работу



**4.** Написать скрипт, подсчитывающий сумму от 1 до 10. 

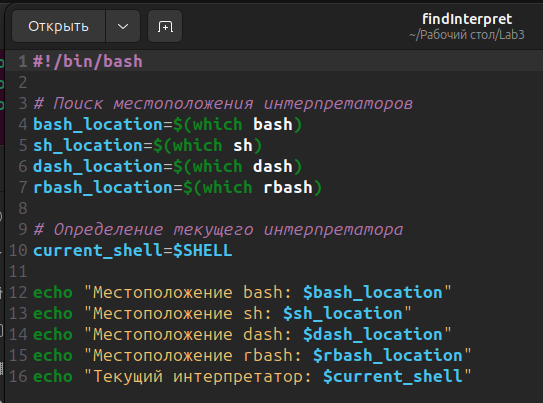


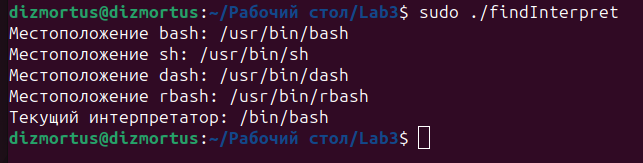
**5** А какие командные интерпретаторы установлены в вашей системе?



sh, bash, rbash, dash

Напишите скрипт поиска местонахождения командных интерпретаторов вашей системы. Также определите какой интерпретатор используется сейчас в вашей системе?





**Задание 2.**

6. Используйте текст первого примера, заменить некоторые фамилии на фамилии членов вашей команды (например, Nutrina заменить на Stankewich).

7. Отсортируйте данные файла так, чтобы он не поменял свое собственное имя. (Решение. Нужно предварительно создать промежуточный отсортированный файл t11.txt. Затем выполнить нужные команды).

8. Подсчитайте число одинаковых слов в обоих файлах. Для подсчета числа слов в файле используйте команду: wc -w file1.

9. Написать скрипт, который создает отсортированный файл, содержащий слова из двух файлов, исключая их общую часть одинаковых слов.

10.Вывести фамилию самого молодого человека (в файле basa.txt – это второй столбец).

11.Вывести зарплату самого молодого человека (зарплата – третий столбец).

12.Вывести отсортированный список имен файлов, начальная часть имени есть script. 13.Написать скрипт, который в каждой строке файла оставляет только буквенные символы, а остальные символы выбрасывает.

Пример. Пусть содержимое файла есть:

Privet year 2022

Good bye year 2021.

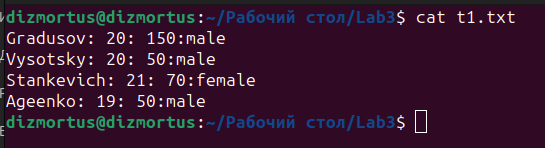
Должны получить такой файл:

Privet year

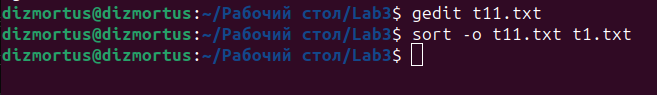
Good bye year

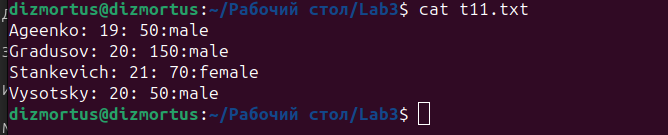
14.Вывести упорядоченный список имен файлов, в именах которых содержится символ подчеркивания, например, script\_sort.

**6.** Используйте текст первого примера, заменить некоторые фамилии на фамилии членов вашей команды (например, Nutrina заменить на Stankewich).

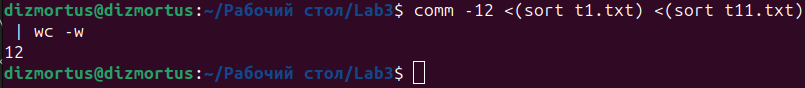


**7.** Отсортируйте данные файла так, чтобы он не поменял свое собственное имя. (Решение. Нужно предварительно создать промежуточный отсортированный файл t11.txt. Затем выполнить нужные команды).



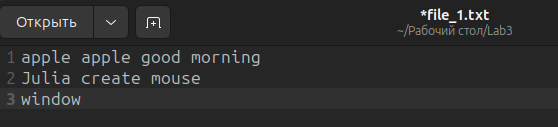


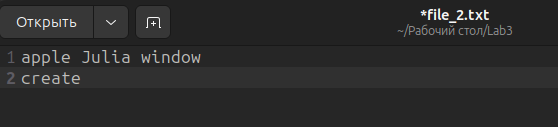
**8.** Подсчитайте число одинаковых слов в обоих файлах. Для подсчета числа слов в файле используйте команду: wc -w file1.



**9.** Написать скрипт, который создает отсортированный файл, содержащий слова из двух файлов, исключая их общую часть одинаковых слов.

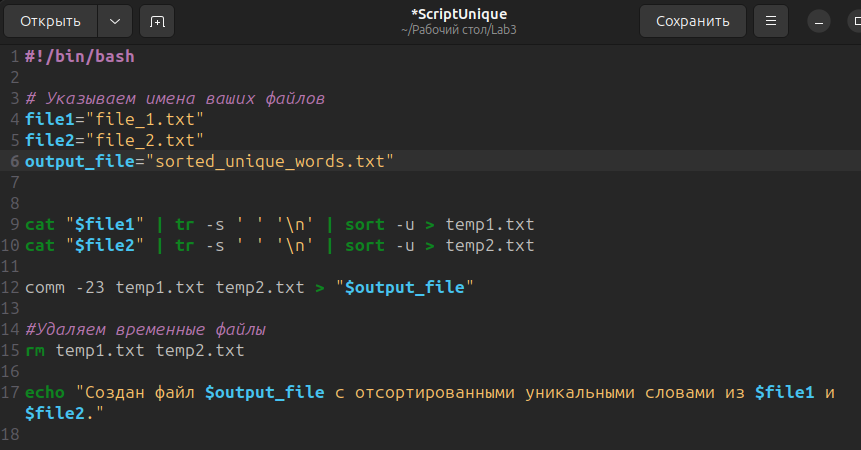
1. Создаем два файла и заполняем их.



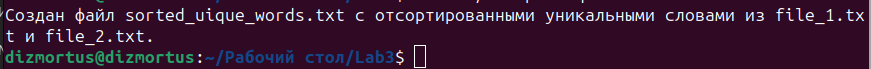


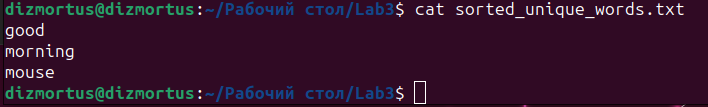
2. Пишем скрипт.

Здесь мы при помощи tr -s разделяем заменяем пробелы, но символ '\n', то есть переводим слова из одной строки на новые. Затем при помощи sort -u сортируем файл и удаляем дубликаты. И записываем результаты в два временных файла. Потом при помощи comm -23 сравниваем содержимое временных файлов и выводим только строки, которые есть только в первом файле (**-2**) и только во втором файле (**-3**).

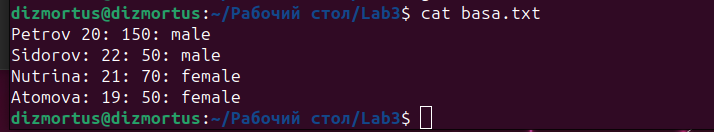


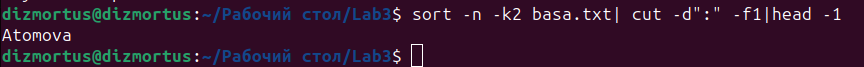
3. Проверяем результат.



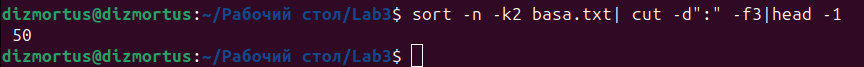


**10.** Вывести фамилию самого молодого человека (в файле basa.txt – это второй столбец).

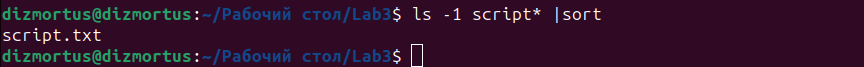




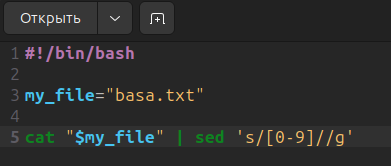
**11.** Вывести зарплату самого молодого человека (зарплата – третий столбец).

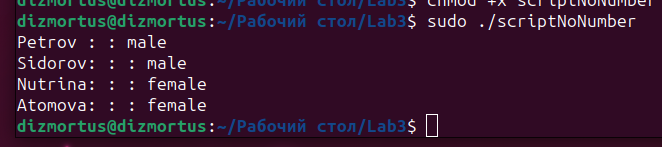


**12.** Вывести отсортированный список имен файлов, начальная часть имени есть script.

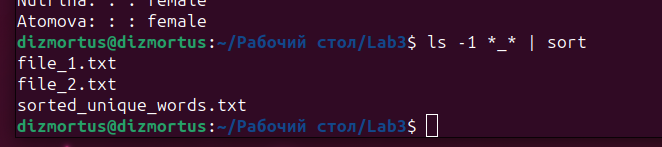


**13.** Написать скрипт, который в каждой строке файла оставляет только буквенные символы, а остальные символы выбрасывает.

****



**14.** Вывести упорядоченный список имен файлов, в именах которых содержится символ подчеркивания, например, script\_sort.

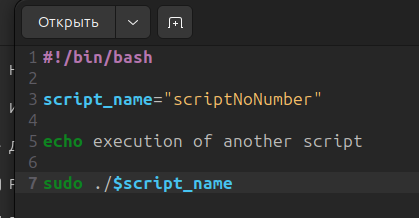


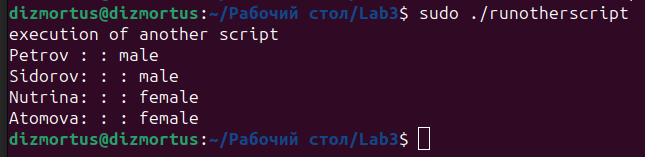
**Задание 3.**

15.Напишите скрипт запуска другого скрипта.

16.Расположите скрипт в одной из стантадртных директорий системы, и запустите его из домашней директории пользователя.

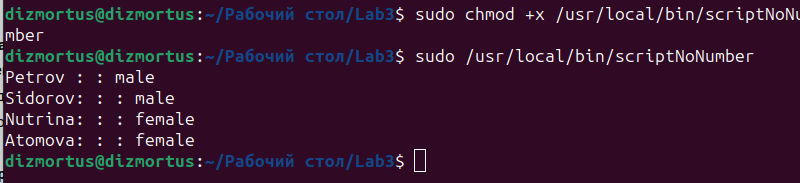
**15.** Напишите скрипт запуска другого скрипта.

****



**16.** Расположите скрипт в одной из стантартных директорий системы, и запустите его из домашней директории пользователя.





Контрольные вопросы:

Управление терминалом

1. В каких технологиях программирования используется умение рабо-

тать с терминалом UNIX?

2. Что такое GUI и CLI и в чем их отличие?

3. Что такое терминал, и чем он отличается от консоли?

4. Укажите преимущества терминала перед оконным интерфейсом, и

наоборот?

5. Что такое командный интерпретатор?

6. Какой командный интерпритатор используется в вашей экосистеме?

7. В какой директории находится командный интерпритатор вашей эко-

системы?

8. Что такое интерфейс командной строки?

9. Как запустить терминал с помощью горячих клавиш?

10. Как запустить консоль и затем вернуться в терминал (GUI)?

11. Что такое команды терминала и в какой директории системы они

находятся?

12. Что такое свойство терминала – автодополнение, и как оно вызыва-

ется (горячие клавиши)?

13. Чем отличаются понятия: «папка» от «директория (каталог)», и есть ли

в терминале «папки»?

14. Как запустить терминал с помощью мыши из текущей папки?

15. Объясните приглашение командной строки (prompt), расшифруйте

его специальные символы.

Навигация по файловой системе

16. Что такое абсолютный и относительный путь к файлу?

17. Что такое корневой каталог, как он называется и обозначается в фай-

ловой системе?

18. Что такое домашний каталог пользователя?

19. Укажите специальные символы текущей и родительской директо-

рии, и как можно сменить текущую директорию?

20. Почему в именах файлов нельзя использовать небуквенные символы

и пробелы?

21. Что такое экранирование символов и имен файлов, и как это дела-

ется?

22. Что такое история введенных команд?

23. Какими средствами UNIX перехватываются и интерпретируются соче-

тания горячих клавиш?

24. В чем заключается мощь командной строки?

25. Что такое регулярные выражения? Опишите интерфейс регулярных

выражений.

26. Каие типы регулярных выражений вы знаете?

Получение справки

27. Что такое справочник man? Как вызвать справку о справке?

28. Назовите основные разделы справочника man.

29. Как проводиться навигация по справочнику man, укажите основные

опции (флаги) навигации?

30. Что такое встроенная справка утилиты, как ее получить?

31. Как получить справку о сочетании горячих клавиш, перехватываемых

терминалом и интерпретируемых командной оболочкой?

Контрольные ответы

1. Умение работать с терминалом UNIX используется в различных технологиях программирования, таких как Bash, Python, Perl, Ruby, и многих других, которые позволяют выполнять команды через командную строку.
2. **GUI** (графический пользовательский интерфейс) и **CLI** (интерфейс командной строки) - это два способа взаимодействия с компьютерной системой. Отличие между ними заключается в том, что GUI использует графические элементы, такие как окна, кнопки и иконки, для взаимодействия, в то время как CLI оперирует текстовыми командами.
3. Терминал - это программа, которая позволяет пользователю взаимодействовать с операционной системой через текстовый интерфейс. Консоль - это физическое устройство (или его эмуляция), на котором можно взаимодействовать с терминалом.
4. Преимущества терминала перед GUI:

* Эффективность при выполнении задач в командной строке.
* Больше возможностей для автоматизации и сценариев.
* Меньшее потребление системных ресурсов. Преимущества GUI:
* Большая наглядность и интуитивность для новичков.
* Удобство для задач, требующих графического взаимодействия.
* Меньший порог входа для пользователей без опыта в командной строке.

1. Командный интерпретатор - это программа, которая выполняет команды, введенные пользователем через интерфейс командной строки. Примеры включают Bash, PowerShell, Command Prompt и другие.
2. Это зависит от операционной системы. Например, в системах Linux и macOS часто используется Bash, а в Windows - Command Prompt или PowerShell.
3. Командный интерпретатор обычно находится в системной директории, например, в Linux это может быть /bin/bash.
4. Интерфейс командной строки - это способ взаимодействия с компьютерной системой, вводя текстовые команды в терминал или консоль.
5. Горячие клавиши для запуска терминала зависят от операционной системы и настроек. Например, в Ubuntu вы можете использовать Ctrl + Alt + T.
6. Для запуска консоли и возврата в GUI, вы можете использовать сочетание Ctrl + Alt + F1 для переключения на текстовую консоль и Ctrl + Alt + F7 (или другую F-клавишу) для возврата в GUI в системах Linux.
7. Команды терминала - это текстовые команды, которые выполняют операции в файловой системе или управляют системой. Они находятся в системных директориях, указанных в переменной PATH.
8. Автодополнение - это функция, которая автоматически предлагает варианты завершения команды или имени файла, когда вы начинаете вводить текст. В большинстве терминалов она вызывается клавишей Tab.
9. Понятия "папка" и "директория (каталог)" синонимичны и используются для обозначения местоположения файлов на диске. В терминале они обозначаются как "директории".

**14.** Для запуска терминала с помощью мыши из текущей папки в Linux, вы можете выполнить следующие действия:

* Использовать контекстное меню файлового менеджера: В большинстве графических файловых менеджеров, таких как Nautilus (в среде GNOME) или Dolphin (в среде KDE), вы можете щелкнуть правой кнопкой мыши на пустом месте в текущей папке и выбрать опцию "Открыть в терминале" или подобную команду.
* Воспользоваться горячей клавишей: На некоторых дистрибутивах Linux можно использовать комбинацию клавиш, такую как Ctrl+Alt+T, чтобы открыть терминал из любого места.

**15.** Приглашение командной строки (prompt) - это текстовый символ или строка, которая отображается в командном интерфейсе и ожидает ввода команды пользователя. Специальные символы в приглашении могут включать:

* $: Обычное приглашение, предполагает работу под обычным пользователем.
* #: Приглашение суперпользователя (root), предполагает выполнение команд с административными правами.
* ~: Обозначение домашней директории пользователя.
* \u: Переменная, заменяемая на имя текущего пользователя.
* \h: Переменная, заменяемая на имя хоста (компьютера).

**16.** Абсолютный путь к файлу полностью определяет местоположение файла в файловой системе, начиная с корневой директории. Он всегда начинается с /. Относительный путь к файлу указывает путь относительно текущей рабочей директории и не начинается с /.

**17.** Корневой каталог в файловой системе Linux называется / и обозначается как "корень". Это самая верхняя директория в иерархии файловой системы, из которой все остальные директории и файлы в системе происходят.

**18.** Домашний каталог пользователя - это директория, предназначенная для хранения личных файлов и настроек пользователя. Обычно он обозначается символом ~ и содержит поддиректорию с именем пользователя, например, /home/имя\_пользователя.

**19.** Специальные символы для текущей и родительской директории:

* .: Текущая директория.
* ..: Родительская директория.

Для смены текущей директории в командной строке Linux вы можете использовать команду cd. Например, cd /путь/к/новой/директории переключит вас в новую директорию по абсолютному пути, а cd ../относительный/путь переключит вас на одну директорию назад.

1. В именах файлов в Linux нельзя использовать небуквенные символы и пробелы из-за того, что это может привести к проблемам с интерпретацией этих имен в командной строке. Небуквенные символы и пробелы могут быть специальными символами, используемыми для разделения аргументов команд и выполнения других задач. Использование этих символов в именах файлов может вызвать некорректное выполнение команд и привести к ошибкам.
2. Экранирование символов в именах файлов означает использование обратной косой черты () перед символом, чтобы указать, что этот символ следует рассматривать как обычный символ, а не как специальный символ. Например, чтобы создать файл с именем "file with spaces.txt", вы можете использовать экранирование пробелов следующим образом: "file\ with\ spaces.txt".
3. История введенных команд в Linux позволяет просматривать и повторно выполнять предыдущие команды. История хранится в файле, обычно в ~/.bash\_history. Вы можете использовать команды, такие как history, чтобы просматривать историю команд, и !!, чтобы выполнить последнюю команду.
4. В UNIX комбинации горячих клавиш могут быть перехвачены и интерпретированы различными программами и оконными менеджерами. Например, в текстовых редакторах такие комбинации могут использоваться для быстрого выполнения действий, таких как сохранение файла или открытие нового. Какие комбинации горячих клавиш доступны и как они интерпретируются, зависит от программы или окружения.
5. Мощь командной строки в том, что она предоставляет множество утилит и команд, которые позволяют пользователям выполнять различные задачи, управлять файлами, настраивать систему и многое другое, всё это без необходимости использования графического интерфейса. Командная строка также позволяет создавать скрипты для автоматизации задач.
6. Регулярные выражения (regex) - это мощный инструмент для поиска и обработки текста на основе шаблонов. Интерфейс регулярных выражений предоставляет набор специальных символов и правил, которые позволяют создавать сложные шаблоны для сопоставления текста. Они широко используются в Linux для поиска, фильтрации и замены текста.
7. В регулярных выражениях существует несколько типов, включая базовые и расширенные. В Linux наиболее часто используются базовые регулярные выражения (BRE) и расширенные регулярные выражения (ERE). BRE использует некоторые основные символы для сопоставления шаблонов, в то время как ERE позволяет использовать более мощные конструкции. Команды, такие как grep и sed, поддерживают регулярные выражения.
8. Справочник man (сокращение от "manual") в Linux и UNIX - это система документации, которая предоставляет информацию о командах, утилитах и других аспектах операционной системы. Справочник man доступен через командную строку и содержит подробные описания команд и их опций.

Чтобы вызвать справку о справке (man по man), можно выполнить команду man man.

1. Основные разделы справочника man обычно делятся на следующие:

* Раздел 1: Команды и утилиты, доступные для обычных пользователей.
* Раздел 2: Системные вызовы (функции, которые использует ядро системы).
* Раздел 3: Библиотечные функции (функции, предоставляемые библиотеками C).
* Раздел 4: Файлы и форматы (описания файловых форматов и структур данных).
* Раздел 5: Стандартные файлы и форматы конфигурации.
* Раздел 6: Игры и развлечения (игры и различные программы).
* Раздел 7: Различные темы (например, макросы форматирования man-страниц).
* Раздел 8: Системное администрирование и управление (команды для администраторов системы).

1. Для навигации по справочнику man используются следующие основные опции (флаги):

* man [раздел] название - позволяет просматривать страницу справки для указанной команды в указанном разделе.
* man -k ключевое\_слово - выполняет поиск команд, содержащих указанное ключевое слово.
* man -f команда - предоставляет краткое описание команды.
* man -S раздел - указывает, из какого раздела брать справку (например, man -S 3 printf).

1. Встроенная справка утилиты обычно доступна через опцию --help или -h. Например, чтобы получить справку по команде ls, вы можете выполнить ls --help.
2. Чтобы получить справку о сочетании горячих клавиш, перехватываемых терминалом и интерпретируемых командной оболочкой, вам следует обратиться к справке для вашей конкретной командной оболочки. Например, для оболочки Bash вы можете использовать команду help или man bash и перейти к соответствующему разделу, который описывает горячие клавиши и функции оболочки.