МИНИСТЕСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники

Отчет по лабораторной работе № 1

на тему «Установка, настройка и использованиедистрибутива Linux»

по дисциплине «Системное программирование»

Кафедра программного обеспечения информационных систем и технологий

Выполнили

Студенты группы 10702122 Н.В. Тимощенко

Д.Е. Кудрявец

Проверил доцент Галай В.В

Минск 2024

**Цель работы:**

Изучить командную строку UNIX и получить основы работы с термина-

лом в дистрибутивах Linux (UNIX).

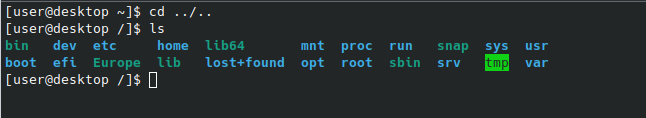
**Выполнение работы:**

**Задание 1:**

Пункт 1:

Команды: cd ../..; ls

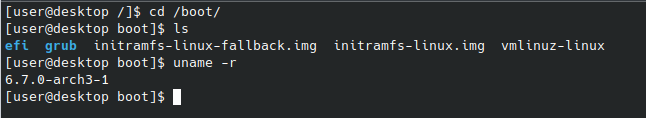
Результат:



Пункт 2:

Команды: cd /boot; uname -r

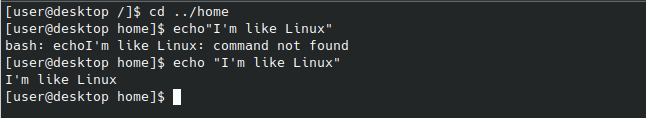
Результат:



Пункт 3:

Команды: cd ../home; echo «I’m lile Linux!»

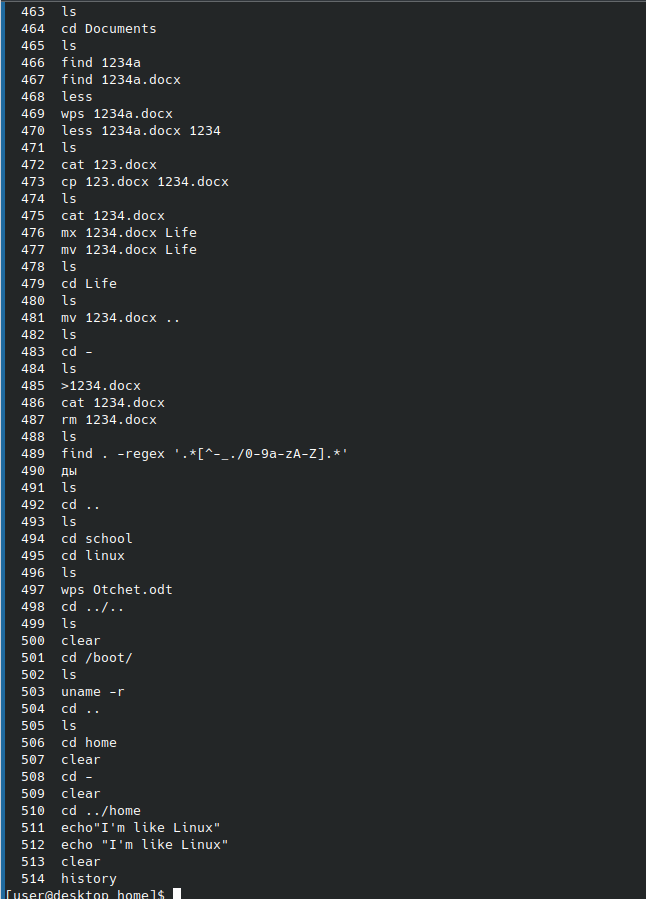
Результат:



Пункт 4:

Команды: history

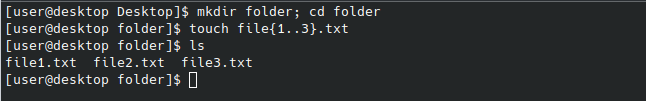
Результат:



Пункт 5:

Команды: cd Desktop; mkdir folder; cd folder; touch file{1..3}.txt; ls

Результат:



Пункт 6:

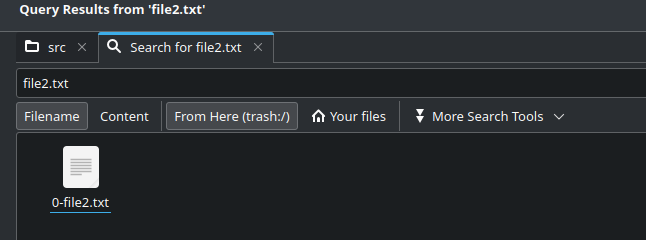
Команды: rm file3.txt

Результат:

Пункт 7:

Команды: trash file2.txt

Результат:

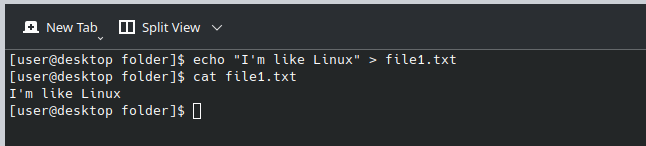


Пункт 8:

Команды: echo "I'm like Linux!" > file1.txt

сat file1.txt

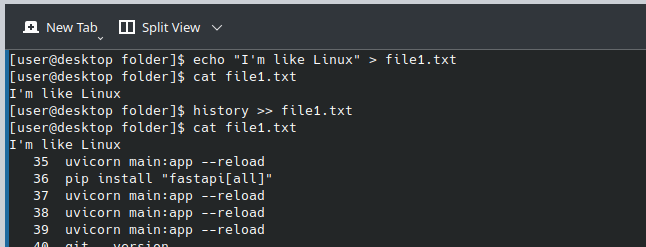
Результат:



Пункт 9-10:

Команды: history >> file1.txt; file1.txt

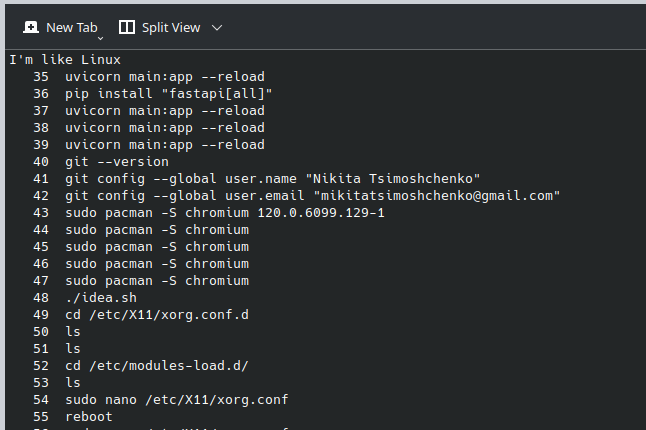
Результат:



Пункт 11:

Команды: vim file1.txt

Результат:

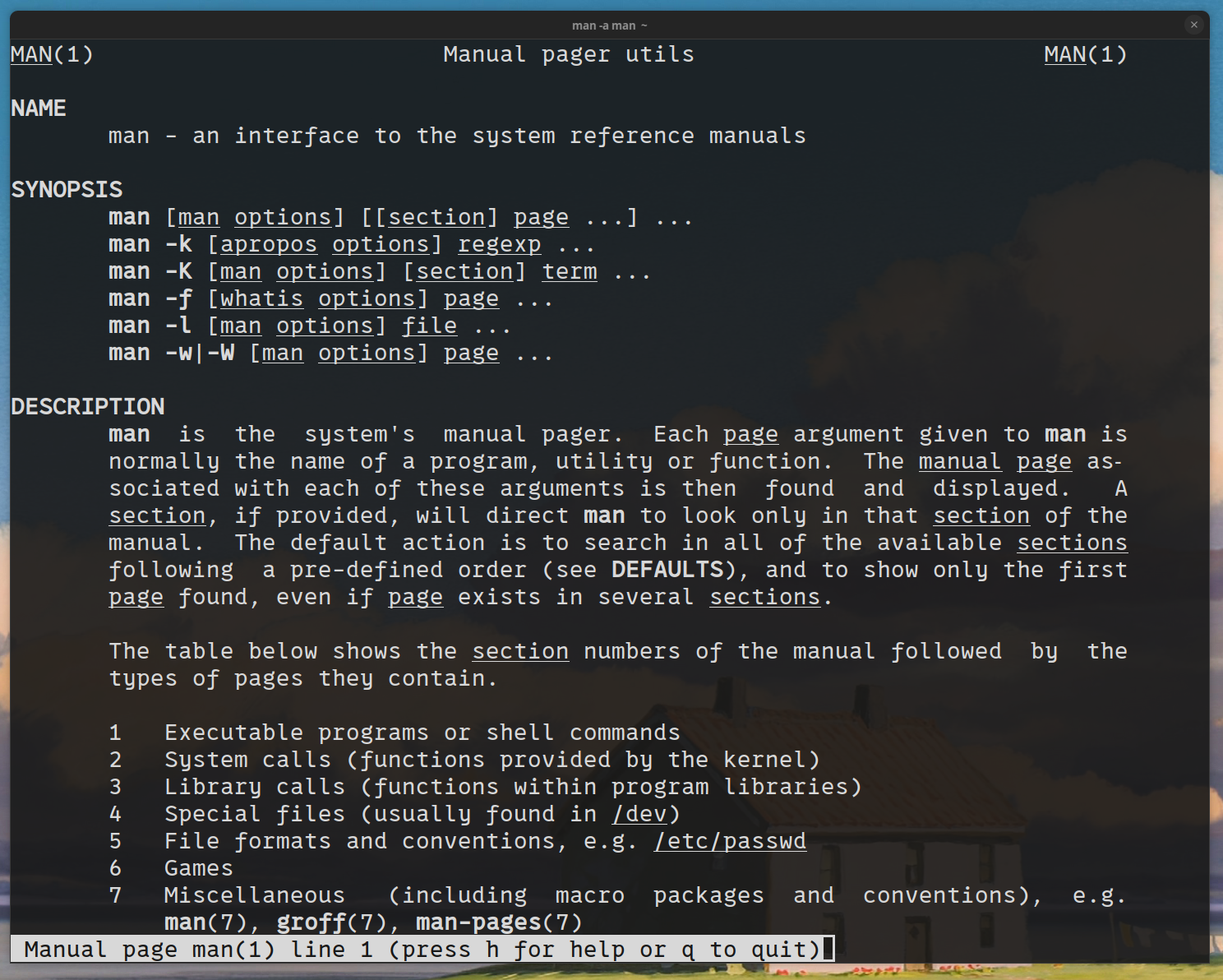
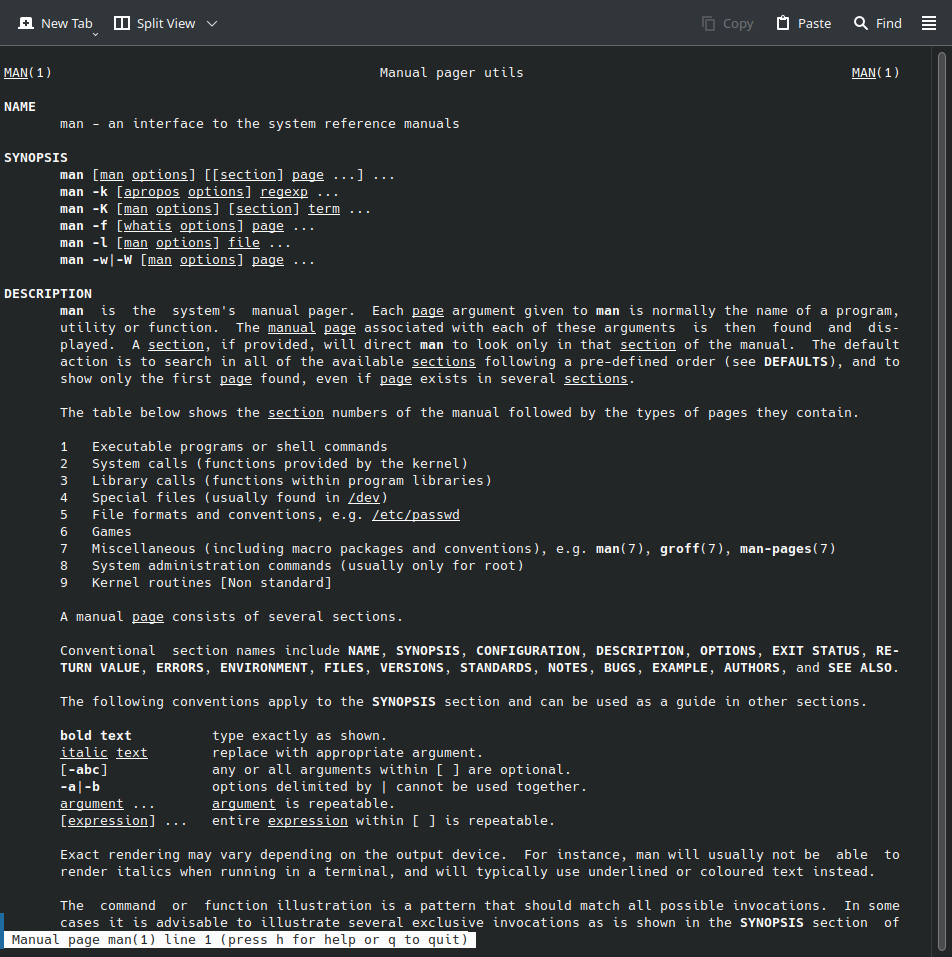


**Задание 2:**

Пункт 1:

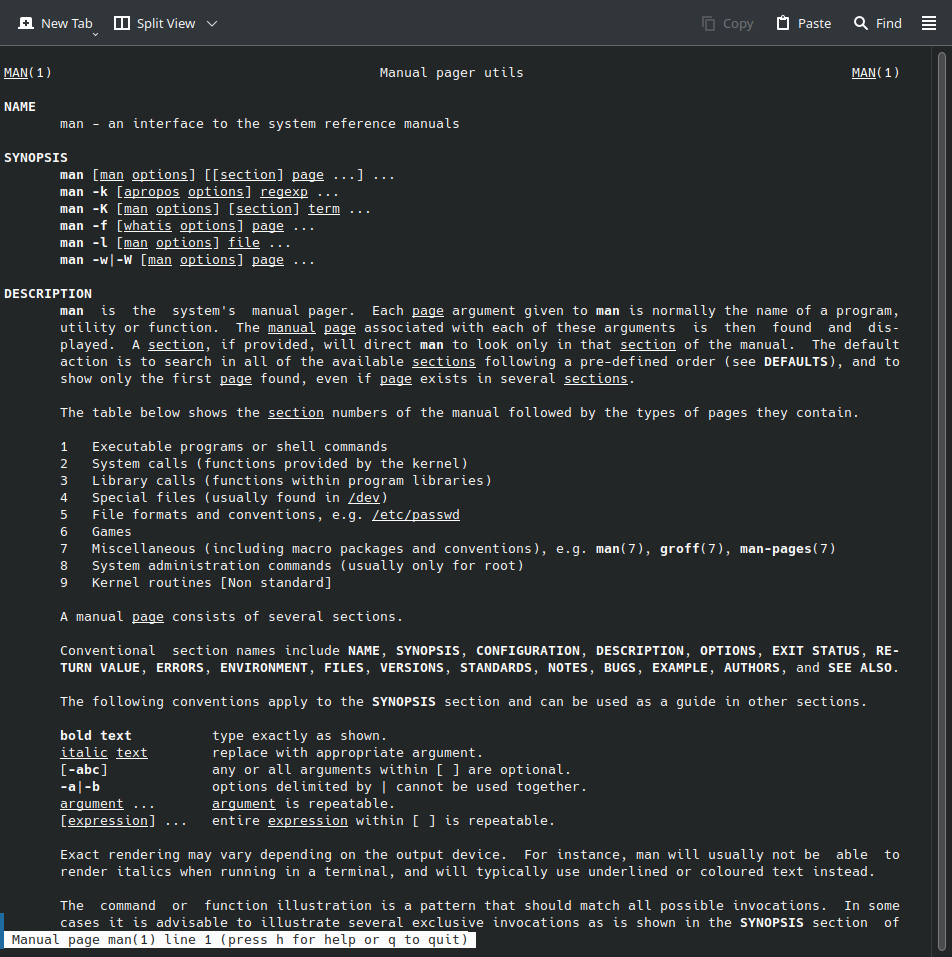
Команды: man -a man

Результат**:**



Пункт 2:

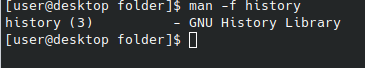
Команды: man 1 man

Результат:

Пункт 3:

Команды: man -f history

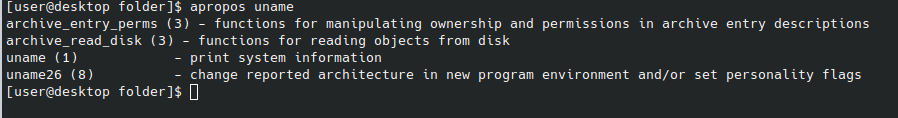
Результат:



Пункт 4:

Команды: apropos uname

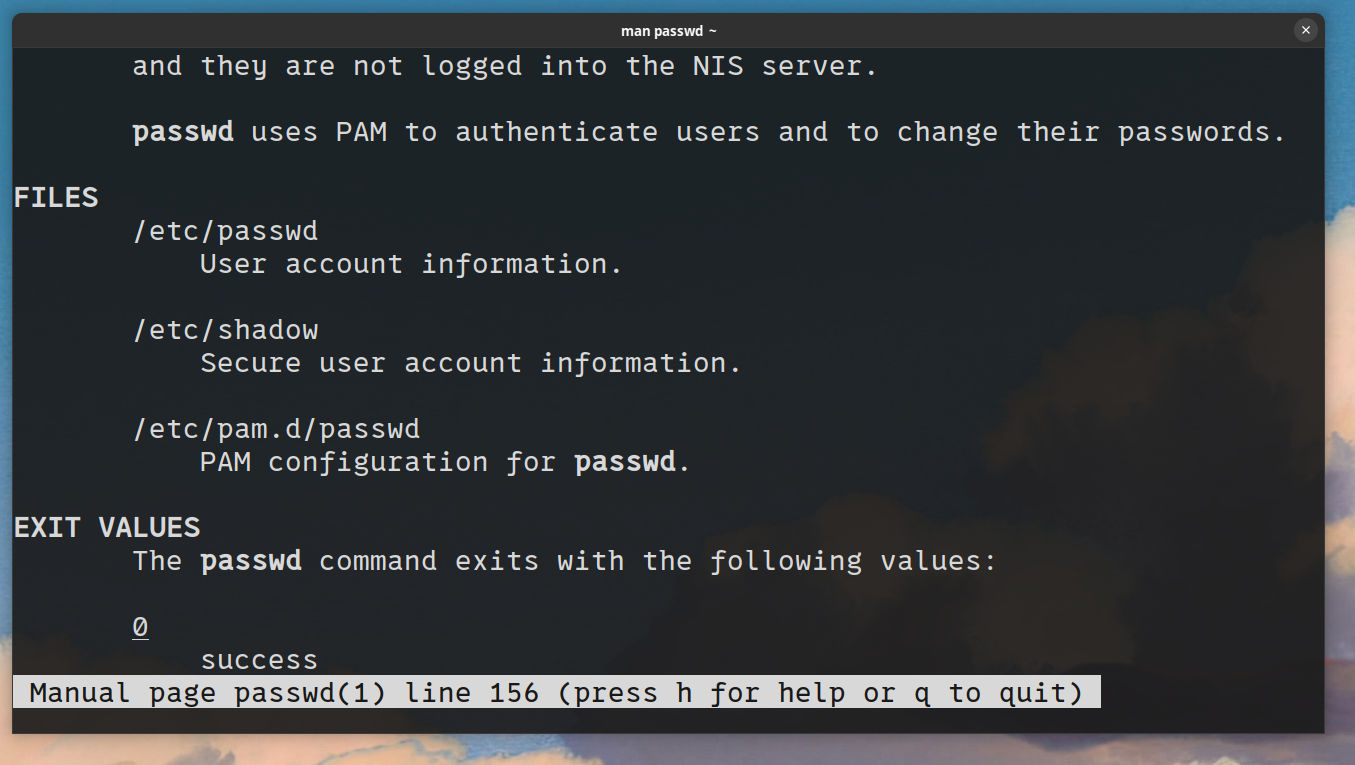
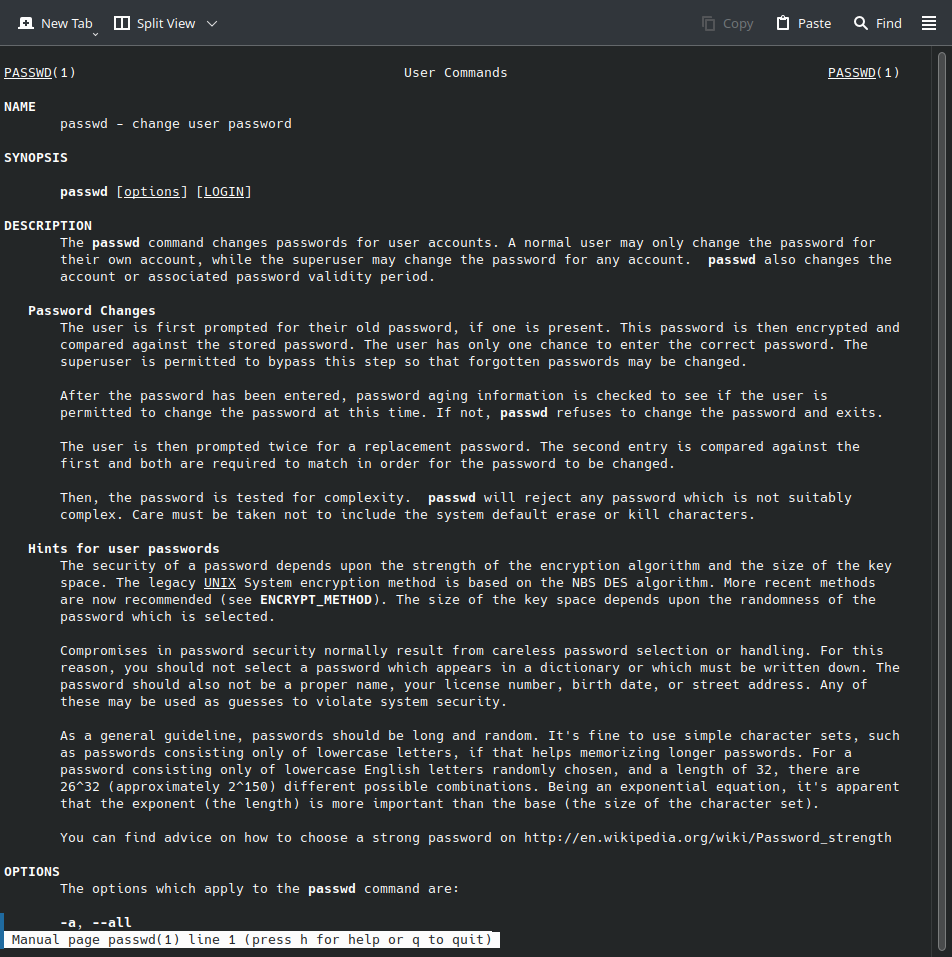
Результат:



Пункт 5:

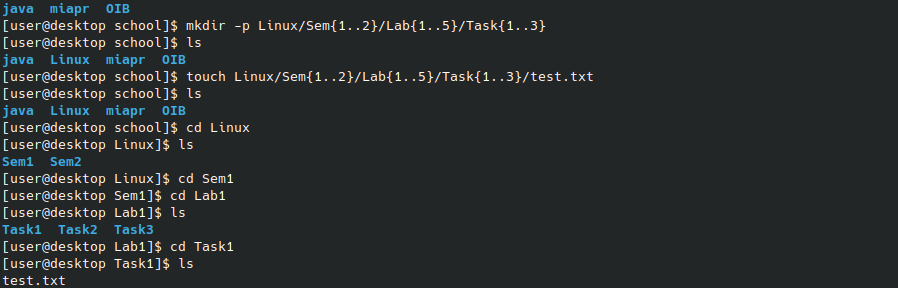
Команды: man passwd

Результат:



**Задание 3.**

**Команда:** mkdir -p Linux/Sem{1..2}/Lab{1..5}/Task{1..3}; touch Linux/Sem{1..2}/Lab{1..5}/Task{1..3}/test.txt

**Результат:** 

**Ответы на контрольные вопросы:**

1. Умение работать с терминалом UNIX используется в следующих технологиях программирования:
   * Разработка серверных приложений: Многие серверные приложения и веб-серверы управляются через командную строку.
   * Администрирование систем: Системные администраторы часто используют терминал для управления серверами и другими системами.
   * \*\*Разработка на языках программирования, таких как Python, Ruby, Perl и других, которые часто используются в UNIX-подобных системах.
   * Работа с системами контроля версий: Git и другие системы контроля версий часто управляются через командную строку.
   * Автоматизация задач: Написание скриптов для автоматизации различных задач, таких как резервное копирование, развертывание приложений и т.д.
   * Работа с контейнерами и оркестрацией: Docker, Kubernetes и другие инструменты для контейнеризации и оркестрации часто управляются через командную строку.
2. GUI (Graphical User Interface) и CLI (Command Line Interface):
   * GUI: Графический пользовательский интерфейс, который позволяет пользователям взаимодействовать с системой через графические элементы, такие как окна, иконки, меню и кнопки. Примеры: Windows, macOS, GNOME, KDE.
   * CLI: Командный интерфейс, который позволяет пользователям взаимодействовать с системой через ввод текстовых команд. Примеры: Bash, PowerShell, cmd.
   * Отличие: GUI предоставляет визуальные элементы для взаимодействия, тогда как CLI требует ввода текстовых команд. GUI обычно более интуитивно понятен для новичков, тогда как CLI может быть более мощным и гибким для опытных пользователей.
3. Терминал и консоль:
   * Терминал: Программное обеспечение, которое эмулирует физический терминал и позволяет пользователю вводить команды и получать вывод от системы. Примеры: Terminal в macOS, GNOME Terminal в Linux.
   * Консоль: Физическое устройство или программное обеспечение, которое предоставляет доступ к командной строке. В некоторых контекстах термины "терминал" и "консоль" используются как синонимы, но исторически консоль может также обозначать физическое устройство.
   * Отличие: Терминал обычно относится к программному обеспечению, тогда как консоль может относиться как к программному, так и к физическому устройству.
4. Преимущества терминала перед оконным интерфейсом и наоборот:
   * Преимущества терминала:
     + Мощность и гибкость: Терминал позволяет выполнять сложные команды и скрипты, которые могут быть трудно или невозможно выполнить через GUI.
     + Автоматизация: Легко автоматизировать задачи с помощью скриптов.
     + Ресурсы: Терминал обычно требует меньше системных ресурсов по сравнению с GUI.
     + Удаленное управление: Легко управлять удаленными серверами через SSH.
   * Преимущества оконного интерфейса:
     + Интуитивность: GUI обычно более интуитивно понятен для новичков.
     + Визуализация: Легче работать с графическими данными и мультимедийными файлами.
     + Многозадачность: Легче управлять несколькими задачами одновременно.
     + Доступность: Большинство пользователей привыкли к GUI, что делает его более доступным для широкой аудитории.
5. Командный интерпретатор:
   * Командный интерпретатор — это программа, которая выполняет команды, введенные пользователем в командной строке. Он интерпретирует команды и выполняет соответствующие действия. Примеры командных интерпретаторов включают Bash (Bourne Again Shell) в UNIX-подобных системах, PowerShell в Windows и cmd в Windows. Командный интерпретатор предоставляет интерфейс для взаимодействия с операционной системой и выполнения различных задач.
6. Какой командный интерпретатор используется в вашей экосистеме?
   * В UNIX-подобных системах (например, Linux и macOS) чаще всего используется Bash (Bourne Again Shell). В Windows могут использоваться PowerShell или cmd.exe.
7. В какой директории находится командный интерпретатор вашей экосистемы?
   * В UNIX-подобных системах Bash обычно находится в /bin/bash.
   * В Windows PowerShell находится в C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe, а cmd.exe находится в C:\Windows\System32\cmd.exe.
8. Что такое интерфейс командной строки?
   * Интерфейс командной строки (CLI) — это текстовый интерфейс, который позволяет пользователям вводить команды в виде текста для управления операционной системой и приложениями. CLI предоставляет прямой доступ к функциям системы и позволяет выполнять сложные задачи с помощью скриптов.
9. Как запустить терминал с помощью горячих клавиш?
   * В разных операционных системах и окружениях рабочего стола горячие клавиши могут различаться:
     + Linux (GNOME): Ctrl + Alt + T
     + macOS: Cmd + Space, затем ввести "Terminal" и нажать Enter
     + Windows (PowerShell): Win + X, затем выбрать "Windows PowerShell" или "Windows Terminal"
     + Windows (cmd.exe): Win + R, затем ввести cmd и нажать Enter
10. Как запустить консоль и затем вернуться в терминал (GUI)?
    * В UNIX-подобных системах консоль и терминал часто используются как синонимы. Если вы имеете в виду переключение между графическим интерфейсом и командной строкой:
      + Linux: Вы можете переключиться между виртуальными консолями с помощью комбинаций клавиш Ctrl + Alt + F1 (для первой виртуальной консоли), Ctrl + Alt + F2 (для второй виртуальной консоли) и так далее. Обычно графический интерфейс находится на Ctrl + Alt + F7 или Ctrl + Alt + F1.
      + macOS: Вы можете запустить Terminal через Spotlight (Cmd + Space, затем ввести "Terminal") и закрыть его, чтобы вернуться в графический интерфейс.
      + Windows: Вы можете запустить cmd.exe или PowerShell через меню "Пуск" или с помощью комбинации клавиш Win + R, затем ввести cmd или powershell и нажать Enter. Закрытие окна командной строки вернет вас в графический интерфейс.
11. Что такое команды терминала и в какой директории системы они находятся?
    * Команды терминала — это текстовые команды, которые выполняются в командной строке для управления операционной системой и приложениями. Примеры команд включают ls (список файлов), cd (смена директории), cp (копирование файлов) и т.д.
    * Директории команд:
      + В UNIX-подобных системах команды обычно находятся в директориях /bin, /sbin, /usr/bin, /usr/sbin, /usr/local/bin и других.
      + В Windows команды могут находиться в директориях C:\Windows\System32, C:\Windows и других.
    * Переменная окружения PATH содержит список директорий, в которых система ищет исполняемые файлы команд. Вы можете просмотреть содержимое PATH с помощью команды echo $PATH в UNIX-подобных системах или echo %PATH% в Windows.
12. Что такое свойство терминала – автодополнение, и как оно вызывается (горячие клавиши)?

* Автодополнение — это функция терминала, которая позволяет пользователю быстро завершать команды, имена файлов и директорий, нажимая определенные клавиши. Это ускоряет работу и уменьшает количество ошибок.
* Горячие клавиши для автодополнения:
  + В большинстве UNIX-подобных систем (например, Linux и macOS) автодополнение вызывается с помощью клавиши Tab.
  + В Windows (PowerShell и cmd.exe) автодополнение также вызывается с помощью клавиши Tab.

1. Чем отличаются понятия: «папка» от «директория (каталог)», и есть ли в терминале «папки»?

* Папка и директория (каталог) — это синонимы, которые обозначают контейнер для хранения файлов и других директорий.
* В терминале используется термин директория или каталог. Термин папка чаще используется в графических интерфейсах (GUI), таких как проводник Windows или Finder в macOS.
* В терминале вы можете работать с директориями с помощью команд, таких как ls (список содержимого директории), cd (смена текущей директории), mkdir (создание новой директории) и т.д.

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

* Linux (GNOME):
  1. Откройте файловый менеджер (например, Nautilus).
  2. Перейдите в нужную папку.
  3. Щелкните правой кнопкой мыши внутри папки и выберите "Открыть в терминале" или аналогичный пункт меню.
* macOS:
  1. Откройте Finder и перейдите в нужную папку.
  2. Перетащите папку в окно Terminal, или используйте сервис "Открыть в Terminal" (если установлен).
* Windows:
  1. Откройте проводник и перейдите в нужную папку.
  2. В адресной строке проводника введите cmd и нажмите Enter, чтобы открыть командную строку в текущей папке.
  3. Альтернативно, вы можете удерживать Shift и щелкнуть правой кнопкой мыши внутри папки, затем выбрать "Открыть окно PowerShell здесь" или "Открыть окно командной строки здесь".

1. Объясните приглашение командной строки (prompt), расшифруйте его специальные символы.

* Приглашение командной строки (prompt) — это текст, который отображается в терминале перед вводом команды. Оно содержит информацию о текущем пользователе, имени хоста и текущей директории.
* Пример приглашения командной строки в Bash:

user@hostname:~/current\_directory$

* Расшифровка специальных символов:
  + user — имя текущего пользователя.
  + hostname — имя хоста (компьютера).
  + ~ — символ, обозначающий домашнюю директорию пользователя.
  + current\_directory — текущая директория, в которой находится пользователь.
  + $ — символ, указывающий на то, что пользователь имеет обычные права (не суперпользователь). Для суперпользователя (root) используется символ #.

1. Что такое абсолютный и относительный путь к файлу?

* Абсолютный путь — это полный путь к файлу или директории, начиная с корневого каталога. Он уникально идентифицирует местоположение файла или директории в файловой системе.
  + Пример абсолютного пути в UNIX-подобных системах: /home/user/documents/file.txt
  + Пример абсолютного пути в Windows: C:\Users\User\Documents\file.txt
* Относительный путь — это путь к файлу или директории, который зависит от текущей рабочей директории. Он указывает на местоположение файла или директории относительно текущей директории.
  + Пример относительного пути в UNIX-подобных системах: documents/file.txt (если текущая директория /home/user)
  + Пример относительного пути в Windows: Documents\file.txt (если текущая директория C:\Users\User)
* Отличие: Абсолютный путь всегда начинается с корневого каталога и является уникальным для всей файловой системы, тогда как относительный путь зависит от текущей рабочей директории и может изменяться в зависимости от контекста.

1. Что такое корневой каталог, как он называется и обозначается в файловой системе?

* Корневой каталог — это верхний уровень иерархии файловой системы, от которого начинаются все другие каталоги и файлы.
* Название и обозначение:
  + В UNIX-подобных системах (например, Linux и macOS) корневой каталог обозначается символом /.
  + В Windows корневой каталог обозначается буквой диска, например, C:\.

1. Что такое домашний каталог пользователя?

* Домашний каталог — это директория, которая принадлежит конкретному пользователю и используется для хранения его личных файлов и настроек.
* В UNIX-подобных системах домашний каталог обычно находится в /home/username, где username — это имя пользователя. Например, для пользователя user домашний каталог будет /home/user.
* В macOS домашний каталог обычно находится в /Users/username.
* В Windows домашний каталог обычно находится в C:\Users\username.

1. Укажите специальные символы текущей и родительской директории, и как можно сменить текущую директорию?

* Специальные символы:
  + Текущая директория: Обозначается символом . (точка).
  + Родительская директория: Обозначается символом .. (две точки).
* Как сменить текущую директорию:
  + В UNIX-подобных системах и Windows для смены текущей директории используется команда cd (change directory).
  + Примеры:
    - Перейти в родительскую директорию: cd ..
    - Перейти в домашний каталог: cd ~ (в UNIX-подобных системах) или cd %HOMEPATH% (в Windows)
    - Перейти в конкретную директорию: cd /path/to/directory (в UNIX-подобных системах) или cd C:\path\to\directory (в Windows)

1. Почему в именах файлов нельзя использовать небуквенные символы и пробелы?

* Небуквенные символы и пробелы могут вызывать проблемы в различных контекстах, включая командную строку, скрипты и программы. Вот несколько причин, почему их использование может быть проблематичным:
  + Синтаксис командной строки: Некоторые символы имеют специальное значение в командной строке. Например, в UNIX-подобных системах символы \*, ?, &, |, ;, <, > и другие имеют специальное значение и могут быть интерпретированы командным интерпретатором как часть команды, а не как часть имени файла.
  + Пробелы: Пробелы могут быть интерпретированы как разделители между аргументами команды. Например, если имя файла содержит пробелы, его нужно заключать в кавычки ("file name.txt") или экранировать пробелы (file\ name.txt), что может усложнить работу.
  + Переносимость: Имена файлов с небуквенными символами и пробелами могут вызывать проблемы при переносе между различными операционными системами и файловыми системами.
  + Скрипты и программы: Некоторые скрипты и программы могут неправильно интерпретировать имена файлов с небуквенными символами и пробелами, что может привести к ошибкам и неожиданному поведению.

1. Что такое экранирование символов и имен файлов, и как это делается?

* Экранирование символов и имен файлов — это процесс, при котором специальные символы или пробелы в именах файлов и командах заключаются в кавычки или предваряются экранирующим символом, чтобы они интерпретировались как обычные символы, а не как специальные команды или разделители.
* Как это делается:
  + Кавычки: Имена файлов или команды, содержащие пробелы или специальные символы, можно заключить в двойные (") или одинарные (') кавычки.
    - Пример: "file name.txt" или 'file name.txt'
  + Экранирующий символ: В UNIX-подобных системах используется обратная косая черта (\) для экранирования специальных символов.
    - Пример: file\ name.txt
  + Примеры:
    - В UNIX-подобных системах:
    - mv "file name.txt" "new file name.txt"

mv file\ name.txt new\ file\ name.txt

* + - В Windows:
    - move "file name.txt" "new file name.txt"

move file^ name.txt new^ file^ name.txt

1. Что такое история введенных команд?

* История введенных команд — это функция командной строки, которая сохраняет и позволяет просматривать и повторно использовать команды, введенные пользователем в предыдущие сессии. Это удобно для повторного выполнения команд, изменения предыдущих команд и анализа истории работы.
* Как использовать:
  + В UNIX-подобных системах:
    - Просмотреть историю команд: history
    - Выполнить команду из истории по номеру: !номер\_команды
    - Поиск команды в истории: Ctrl + R, затем ввести часть команды
  + В Windows (PowerShell):
    - Просмотреть историю команд: Get-History
    - Выполнить команду из истории по номеру: Invoke-History номер\_команды

1. Какими средствами UNIX перехватываются и интерпретируются сочетания горячих клавиш?

* В UNIX-подобных системах сочетания горячих клавиш перехватываются и интерпретируются с помощью различных утилит и библиотек, таких как:
  + Readline: Библиотека, используемая в Bash и других командных интерпретаторах для обработки ввода пользователя. Она поддерживает множество горячих клавиш для редактирования командной строки, таких как Ctrl + A (перейти в начало строки), Ctrl + E (перейти в конец строки), Ctrl + K (удалить текст от курсора до конца строки) и другие.
  + Terminfo и Termcap: Базы данных, содержащие информацию о возможностях терминалов, включая поддержку горячих клавиш.
  + X Window System: В графических окружениях UNIX-подобных систем горячие клавиши могут перехватываться и интерпретируться с помощью X Window System и связанных с ним библиотек, таких как Xlib.

1. В чем заключается мощь командной строки?

* Мощь командной строки заключается в следующих аспектах:
  + Гибкость и автоматизация: Командная строка позволяет выполнять сложные задачи с помощью скриптов и автоматизировать повторяющиеся задачи.
  + Эффективность: Командная строка позволяет быстро выполнять команды и комбинировать их для достижения сложных целей.
  + Мощные инструменты: Командная строка предоставляет доступ к мощным инструментам и утилитам, таким как grep, awk, sed, find, которые позволяют выполнять сложные операции с данными.
  + Удаленное управление: Командная строка позволяет управлять удаленными серверами и системами через SSH и другие протоколы.
  + Минимальные ресурсы: Командная строка требует меньше системных ресурсов по сравнению с графическими интерфейсами.
  + Программируемость: Командная строка позволяет использовать языки программирования и скрипты для выполнения сложных задач и автоматизации процессов.

1. Что такое регулярные выражения? Опишите интерфейс регулярных выражений.

* Регулярные выражения — это последовательности символов, которые формируют шаблон поиска. Они используются для поиска, замены и манипуляции строками на основе заданных шаблонов.
* Интерфейс регулярных выражений:
  + Синтаксис: Регулярные выражения состоят из литералов (буквы и цифры) и специальных символов, которые обозначают шаблоны поиска.
    - Примеры специальных символов:
      * . — любой символ.
      * \* — ноль или более предыдущих символов.
      * + — один или более предыдущих символов.
      * ? — ноль или один предыдущий символ.
      * [] — набор символов (например, [abc] соответствует любому из символов a, b или c).
      * | — логическое ИЛИ.
      * ^ — начало строки.
      * $ — конец строки.
      * \d — любая цифра.
      * \w — любая буква или цифра.
      * \s — любой пробельный символ.

1. Какие типы регулярных выражений вы знаете?

* Регулярные выражения могут быть классифицированы по нескольким типам в зависимости от их синтаксиса и поддерживаемых функций:
  + POSIX (Portable Operating System Interface) регулярные выражения:
    - POSIX Basic Regular Expressions (BRE): Используются в таких утилитах, как grep и sed. Поддерживают базовый набор символов и операторов.
    - POSIX Extended Regular Expressions (ERE): Используются в таких утилитах, как egrep и awk. Поддерживают более расширенный набор символов и операторов.
  + Perl-совместимые регулярные выражения (PCRE):
    - Используются в языке программирования Perl и многих других языках и инструментах, таких как PHP, Python (через библиотеку re), и другие. Поддерживают очень мощные и гибкие возможности.
  + ECMAScript регулярные выражения:
    - Используются в JavaScript и других языках, поддерживающих стандарт ECMAScript. Поддерживают большинство функций, доступных в PCRE.
  + GNU регулярные выражения:
    - Используются в утилитах GNU, таких как grep, sed, и awk. Поддерживают расширенные возможности, включая POSIX BRE и ERE.

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

* Справочник man — это система документации в UNIX-подобных операционных системах, которая предоставляет справку о командах, системных вызовах, библиотечных функциях и других компонентах системы.
* Как вызвать справку о справке:
  + Для получения справки о самом справочнике man используйте команду:

man man

1. Назовите основные разделы справочника man.

* Основные разделы справочника man включают:
  1. Раздел 1: Исполняемые программы или команды оболочки.
  2. Раздел 2: Системные вызовы (функции ядра).
  3. Раздел 3: Библиотечные вызовы (функции библиотек).
  4. Раздел 4: Специальные файлы (обычно находятся в /dev).
  5. Раздел 5: Форматы файлов и соглашения.
  6. Раздел 6: Игры.
  7. Раздел 7: Различные (включая макросы и соглашения).
  8. Раздел 8: Команды системного администрирования.

1. Как проводится навигация по справочнику man, укажите основные опции (флаги) навигации?

* Навигация по справочнику man:
  + Просмотр страницы справки:

man команда

* + Просмотр страницы справки из определенного раздела:

man раздел команда

Пример: man 2 open для просмотра системного вызова open из раздела 2.

* + Поиск по страницам справки:

man -k ключевое\_слово

или

apropos ключевое\_слово

* + Основные клавиши навигации внутри страницы справки:
    - Прокрутка вниз: Space или f
    - Прокрутка вверх: b
    - Прокрутка вниз на одну строку: Enter или j
    - Прокрутка вверх на одну строку: k
    - Переход к началу страницы: g
    - Переход к концу страницы: G
    - Поиск вперед: / затем ввести строку поиска и нажать Enter
    - Поиск назад: ? затем ввести строку поиска и нажать Enter
    - Выход из справки: q

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

* Встроенная справка утилиты — это справка, которая предоставляется самой утилитой и может быть вызвана с помощью определенных опций или флагов.
* Как получить встроенную справку:
  + Многие утилиты поддерживают опцию --help или -h для вывода краткой справки. Пример:

команда --help

или

команда -h

1. Как получить справку о сочетании горячих клавиш, перехватываемых терминалом и интерпретируемых командной оболочкой?

* Справка о сочетании горячих клавиш, перехватываемых терминалом и интерпретируемых командной оболочкой:
  + Для Bash:
    - Справка о сочетаниях горячих клавиш, поддерживаемых библиотекой Readline, может быть найдена в справочнике man bash или man readline.
    - Пример:

man bash

или

man readline

* + Для других командных оболочек:
    - Справка о сочетаниях горячих клавиш может быть найдена в соответствующей документации или справочнике для этой оболочки.
    - Пример для Zsh:

man zsh

* + Для Windows (PowerShell):
    - Справка о сочетаниях горячих клавиш может быть найдена в официальной документации PowerShell или с помощью команды Get-Help.
    - Пример:

Get-Help about\_Key\_Bindings

Эти команды и инструменты помогут вам получить необходимую информацию о сочетаниях горячих клавиш и их использовании в различных командных оболочках.