**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

****

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

|  |  |
| --- | --- |
| Институт | компьютерных наук |
| Кафедра | автоматизированных систем управления |

ЛАБОРАТОРНАЯ РАБОТА №2

По дисциплине "Операционные системы Linux"

На тему "Работа с файловой системой ОС Linux"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ПИ-22-1 | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | Пахомов А.А. |
| Руководитель | |  |  | Кургасов В.В. |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата |

Липецк, 2024 г.

Оглавление

[Цель работы 3](#_Toc179896011)

[Ход работы 4](#_Toc179896012)

[1.Общая часть 4](#_Toc179896013)

[2.Файлы и каталоги 13](#_Toc179896014)

[3. Пользователи и группы 16](#_Toc179896015)

[4. Архивация и поиск 18](#_Toc179896016)

[5. Создание демона 20](#_Toc179896017)

[Вывод 22](#_Toc179896018)

[Контрольные вопросы 23](#_Toc179896019)

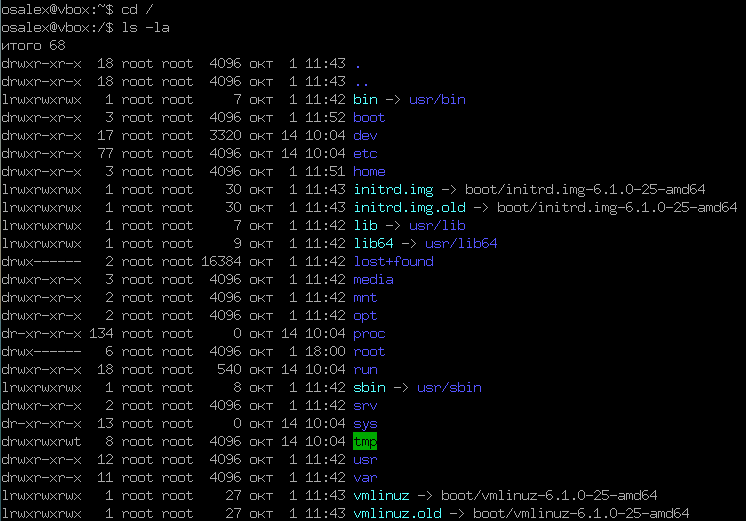
**Цель работы**

Приобрести опыт работы с файлами и каталогами в ОС Linux, настройки прав на доступ к файлам и каталогам.

# **Ход работы**

1. Общая часть
   1. Изучить и привести перечень основных каталогов с указанием их назначения.

Чтобы изучить перечень основных каталогов перейдём с помощью команды cd в корневую директорию «/» и с помощью команды ls выведем на экран все каталоги (см. рис.1)

**

*Рисунок 1 – Каталоги в корневой директории*

Описание назначения всех каталогов представлено в таблице 1.

Таблица 1 – Каталоги и их назначения

|  |  |
| --- | --- |
| **Каталог** | **Назначение** |
| bin | Этот каталог содержит исполняемые файлы. |
| boot | Содержит все файлы, связанные с загрузчиком системы. Это ядро vmlinuz и сами файлы загрузчика. |
| dev | Каталог содержит специальные файлы устройств для всех устройств. |
| etc | В этой папке содержатся конфигурационные файлы всех программ, установленных в системе. |
| home | Содержит домашние каталоги всех пользователей системы и их личные файлы. |
| initrd.img | Содержит временную корневую файловую систему, которая монтируется в процессе загрузки системы в оперативную память. |
| initrd.img.old | Предыдущая версия initrd.img. |
| lib | Содержит файлы системных библиотек, которые используются исполняемыми файлами. |
| lib64 | Хранит варианты библиотек для систем, поддерживающих несколько форматов (x86, x64). |
| lost+found | В папке помещаются фрагменты данных, на которые нет ссылок нигде в файловой системе. |
| media | В этот каталог система монтирует все подключаемые внешние накопители - USB флешки, оптические диски и другие носители информации. |
| mnt | В этот каталог системные администраторы могут монтировать внешние или дополнительные файловые системы. |
| opt | В эту папку устанавливаются проприетарные программы, игры или драйвера. |
| proc | Виртуальная файловая система, которая содержит информацию о запущенных процессах и настройках ядра в виде файлов. |
| root | Содержит личные файлы администратора системы. |
| run | Каталог, содержащий PID файлы процессов |
| sbin | Содержит системные программы. |
| srv | В этом каталоге содержатся файлы серверов и сервисов. |
| sys | Содержит информацию об устройствах, драйверах и некоторых свойствах ядра. |
| tmp | В этом каталоге содержатся временные файлы, созданные системой, любыми программами или пользователями. |
| usr | В этой директории находятся исполняемые файлы, исходники программ, различные ресурсы приложений. |
| var | Содержит переменные данные, обрабатываемые демонами. Включает в себя файлы, которые часто изменяются. |
| vmlinuz | Сжатый образ ядра Linux, который загружается при старте системы. |
| vmlinuz.old | Предыдущая версия ядра. |

1.2. Зайти в терминал под root.

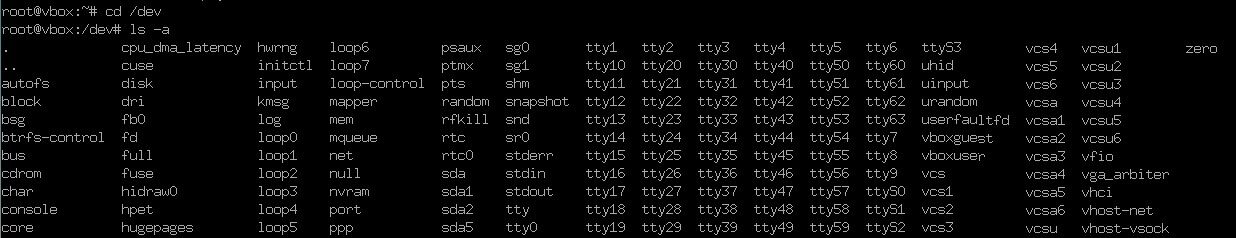
Для получения прав администратора используется команда su (см. рис.2), после которой нужно ввести пароль администратора.

**

*Рисунок 2 – Получение root*

* 1. Посмотреть содержимое каталога файлов физических устройств. В отчете привести перечень файлов физических устройств на рабочем месте с указанием назначения файлов.

Просмотрим содержимое каталога файлов физических устройств с помощью команды ls. Результат работы команды представлен на рисунке 3.

**

*Рисунок 3 – Содержимое директории /dev*

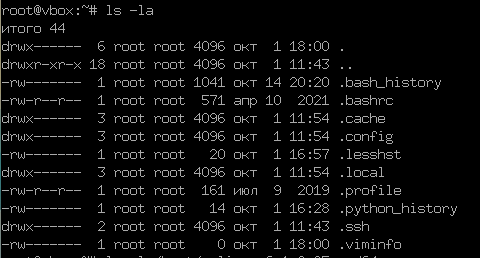
Описание некоторых каталогов из данной директории представлено в таблице 2.

Таблица 2 – Перечень файлов устройств

|  |  |
| --- | --- |
| **Устройство** | **Назначение** |
| autofs | Используется для локального монтирования удалённых директорий |
| bus | Этот файл представляет первый из установленных в системе CD-приводов |
| console | Устройство, на которое должны отправляться системные сообщения и будет проходить авторизация в режиме единственного пользователя. |
| dri | Интерфейс, позволяющий пользовательским приложениями получать прямой доступ к видеоаппаратуре. |
| fb0 | Первый кадровый буфер (абстрактная прослойка между ПО и графическим оборудованием). |
| fd | Привод для дискет. |
| full | Специальный файл, являющийся "полным устройством" – запись в него гарантированно вызывает ошибку "недостатка места". Используется для проверки на отсутствие вывода программы. |
| loop0-7 | Позволят монтировать файлы как изолированные файловые системы, отображая файл как виртуальное блочное устройство. |
| mem | Специальный файл, являющийся образом физической памяти компьютера. |
| null | Пустой файл, обнуляющий любые записанные в него байты. |
| ptmx | Хранит файл мультиплексорного устройства псевдотерминала. Он используется для создания пары основного и подчинённого псевдотерминала. |
| pts | Специальный виртуальный каталог временных файлов. |
| random | Специальное символьное псевдоустройство, предоставляющее интерфейс к системному генератору случайных чисел. |
| urandom | Специальное символьное псевдоустройство, предоставляющее интерфейс к системному генератору псевдослучайных чисел. |
| rfkill | Интерфейс управления радиопередатчиками (WiFi, Bluetooth и прочие). |
| rtc | Интерфейс к драйверам часов реального времени. |
| sda | Специальный файл устройства, предоставляющий доступ к жёстким дискам с SCSI-интерфейсом. Буква в конце обозначает новое устройство, число обозначает номер раздела. |
| sr0 | Приводы оптических дисков. |
| stderr | Хранит стандартный поток ошибок. |
| stdin | Хранит стандартный поток ввода. |
| stdout | Хранит стандартный вывод. |
| tty | Файлы поддержки пользовательских консолей. |
| ttyS | Файлы, обеспечивающие работу с последовательными портами. |
| zero | Специальный файл, являющийся источником нулевых байтов. |

1.4. Перейти в директории пользователя root. Посмотреть содержимое каталога. Посмотреть содержимое файла vmlinuz. Посмотреть и пояснить права доступа к файлу vmlinuz.

Как в пункте 1.2, переходим в администратора и в его домашнюю директорию. На рисунке 4 представлены файлы домашней директории администратора.

**

*Рисунок 4 – Домашняя директория администратора*

Чтобы узнать права доступа ядра используем команду ls -l /boot/vmlinuz-6.1.0-25-amd64 (см. рис. 5).

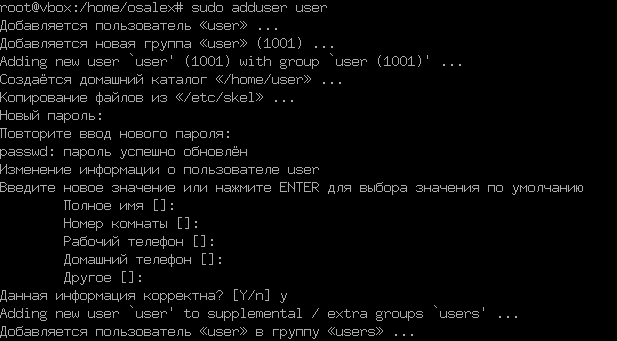
**

*Рисунок 5 – Права доступа на vmlinuz*

Из прав доступа на рисунке 5 видно, что файл доступен для чтения всем пользователям, а администратор (root) имеет право на запись.

1.5. Создать нового пользователя user.

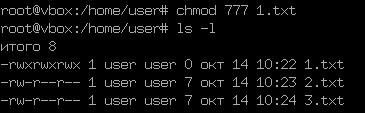
С помощью команды adduser создадим нового пользователя user (см. рис. 6).

**

*Рисунок 6 – Создание нового пользователя*

1.6. Создать в директории пользователя user три файла 1.txt, 2.txt., 3.txt., используя команды touch, cat и тактовый редактор (на выбор vi/nano). Просмотреть и пояснить права доступа к файлам. Изменить права доступа на файл 1.txt в директории пользователя user.

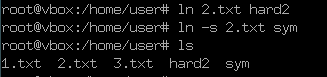
Создадим с помощью текстовых редакторов три файла (1.txt, 2.txt, 3.txt) и с помощью команды chmod поменяем права доступа к первому файлу (см. рис.7). Таким образом, после выполнения команды первый файл могут просматривать, редактировать и исполнять администратор, владелец и пользователи, состоящие в группе.

**

*Рисунок 7 – Изменение прав доступа*

1.7. Создать жесткую и символическую ссылки на файл 2.txt. Просмотреть и описать полученные результаты.

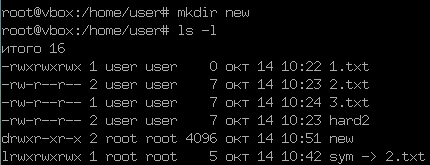
Создадим с помощью команды ln жёсткую ссылку на второй файл и с помощью команды ln -s символическую (см. рис. 8). Жесткая ссылка указывает на те же данные, хранимые на диске, что и оригинальный файл. Символическая ссылка просто хранит путь к файлу.

**

*Рисунок 8 – Создание ссылок на файл*

1.8. Создать каталог new в каталоге пользователя user.

Для создания каталога используется команда mkdir (см. рис. 9)

**

*Рисунок 9 – Создание каталога*

1.9. Скопировать файл 1.txt в каталог new.

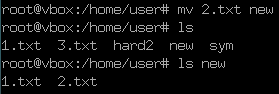
Для копирования файла используем команду cp (см. рис. 10)

**

*Рисунок 10 – Копирование файла*

1.10. Переместить файл 2.txt в каталог new.

Для перемещения файла используем команду mv (см. рис. 11). После этого символическая ссылка перестаёт работать.

**

*Рисунок 11 – Перемещение файла*

1.11. Изменить владельца файла 3.txt и каталога new.

Изменение владельца происходит с помощью команды chown, как показано на рисунке 12.

**

*Рисунок 12 – Изменение владельца файла*

1.12. Удалить файл 1.txt в каталоге new.

Удаление файла происходит благодаря команде rm (см. рис. 13)

**

*Рисунок 13 – Удаление файла*

1.13. Удалить каталог new.

Удаление каталога происходит благодаря команде rm -r (см. рис. 14), означающая рекурсивное удаление каталога.

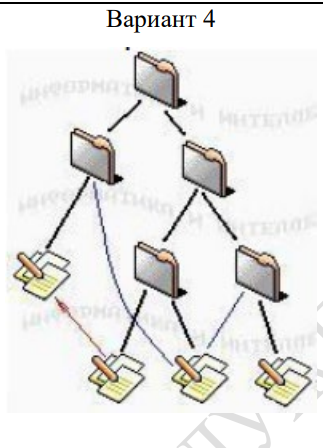
**

*Рисунок 13 – Удаление непустой директории*

1. Файлы и каталоги

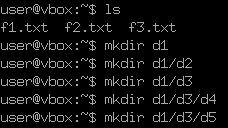
2.1. Создать структуру каталогов в соответствии с вариантом.

Черными линиями представлена вложенность файлов/подкаталогов в каталоги. Синими линиями представлены ссылки. Красными линиями – символические ссылки. Стрелка на красной линии указывает на целевой файл ссылки. Файлы создаются копированием ранее созданных файлов командос cp с внесением в копию некоторых изменений. Ссылки создаются командой ln, символические ссылки – ей же, но с ключом -s. На рисунке 14 представлено задание, соответствующее варианту.

**

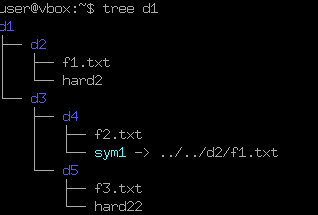
*Рисунок 14 – Структура каталогов*

Создадим с помощью текстовых редакторов vi, mc, tee три текстовых файла, содержащие стихотворения. Затем с помощью команды mkdir создадим структура каталогов в соответствии с вариантом. На рисунке 15 описаны эти действия.

**

*Рисунок 15 – Создание структуры каталогов*

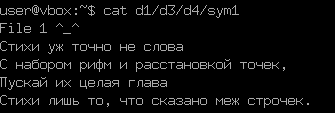
С помощью команды mv переместим файлы в нужные директории. Затем создадим ссылки в соответствии с вариантом с помощью ln. Чтобы проверить результат наглядно используем команду tree (см. рис. 16).

**

*Рисунок 16 – Полученная файловая структура*

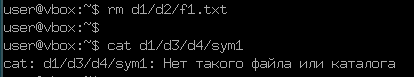
2.2. Провести ряд экспериментов, иллюстрирующих доступ к файлам по основным именам, по ссылкам и по символическим ссылкам.

Проверим, как работают ссылки на файлы. Выведем содержимое первого файла с помощью символической ссылки. Результат показан на рисунке 17.

**

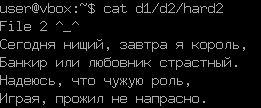
*Рисунок 17 – Вывод содержимого первого файла*

Теперь удалим первый файл и попробуем также по символической ссылке вывести содержимое файла. Как видно из рисунка 18, команда выдала ошибку об отсутствии данного файла.

**

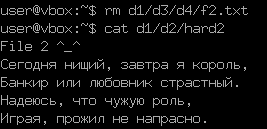
*Рисунок 18 – Ошибка вывода содержимого первого файла*

Сделаем аналогично со вторым файлом, только в этом случае используя жёсткую ссылку (см. рис. 19).

**

*Рисунок 19 – Вывод содержимого второго файла*

При удалении второго файла жёсткая ссылка также продолжает работать, что видно из рисунка 20.

**

*Рисунок 20 – Вывод содержимого второго файла после удаления*

Для удаления всех файлов используем команду rm -r для директории d1.

**3.** **Пользователи и группы**

3.1. Создать пользователей с именем <ВашеИмяГруппа> и <ФамимлияИмяОтца>.

Для создания новых пользователей используем команду adduser. Результат создания новых пользователей представлен на рисунке 21.

**

*Рисунок 21 – Пользователи системы*

3.2. Войти в систему под созданным пользователем <ВашеИмяГруппа>.

Сменить пользователя можно с помощью команды su <username> (см. рис. 22).

**

*Рисунок 22 – Смена пользователя*

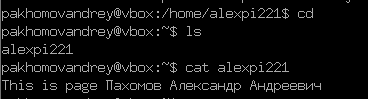
3.3. Работа с файлом

Создаем файл alexpi221 в домашней директории пользователя alexpi221 (см. рис. 23).

**

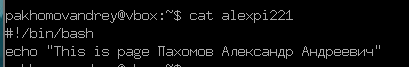
*Рисунок 23 – Содержимое файла*

После добавления текущего пользователя в группу sudo, с помощью команды mv перемещаем файл в домашнюю директорию пользователя pakhomovandrey. На рисунке 24 проверяем, что файл оказался в нужном месте.

**

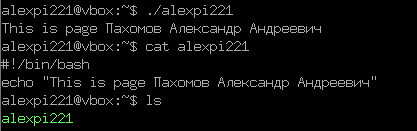
*Рисунок 24 – домашняя директория пользователя pakhomovandrey*

Изменим содержимое файла (см. рис. 25).

**

*Рисунок 25 – Редактирование файла*

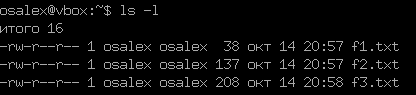
Затем перемещаем обратно файл в директорию alexpi221 и с помощью команды chmod делаем файл исполняемым (см. рис. 26).

**

*Рисунок 26 – запуск исполняемого файла*

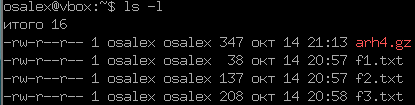
**4.** **Архивация и поиск**

Создадим три текстовых файла в домашней директории (см. рис. 27). tar -czvf arh4.gz file1.txt file2.txt file3.txt

**

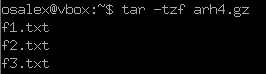
*Рисунок 27 – Домашняя директория*

Затем с помощью команды tar -czvf arh4.gz file1.txt file2.txt file3.txt, создадим архив, содержащий три файла. Флаг «c» означает создание файла, «v» - выводит информацию о процессе, «f» выводит результат в файл. Рассмотрим на рисунке 28, как уменьшился суммарный размер файлов. Проанализируя размеры файлов, получим, что размер файлов уменьшился примерно на 10 процентов.

**

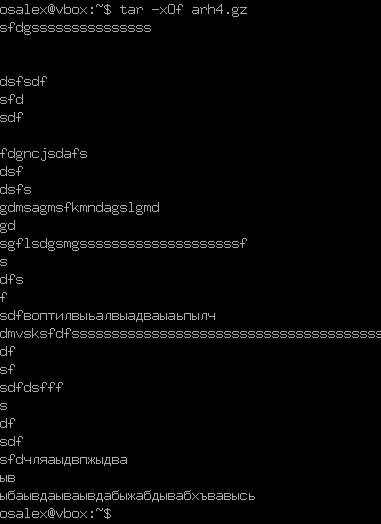
*Рисунок 28 – размер файлов*

Для вывода содержимого архива используем команду tar с флагом «t» (см. рис. 29).

**

*Рисунок 29 – Содержимое архивов*

Для просмотра содержимого конкретного файла архива можно добавить опцию -O (данная опция указывает на вывод содержимого файла на стандартный вывод, то есть в терминал, вместо записи и сохранения в файл). Результат работы программы представлен на рисунке 30.

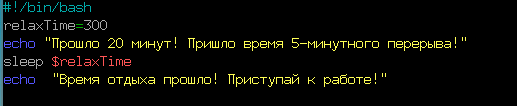
**

*Рисунок 30 – Вывод содержимого файла, расположенного в архиве*

**5. Создание демона**

Daemon – это программа, которая запускается в фоновом режиме автоматически (то есть без взаимодействия с терминалом или пользовательским интерфейсом).

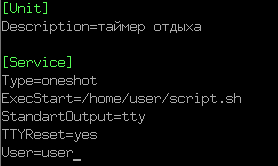
Наша программа представляет собой программу, сообщающую информацию о том, что прошло 20 минут и пользователю следует отдохнуть 5 минут. Код прогhаммы на bash представлена на рисунке 31.

**

*Рисунок 31 – Программа на bash*

После сохранения файла делаем файл исполняемым с помощью команды chmod.

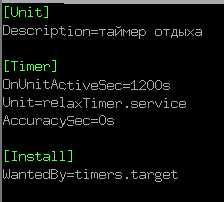
Затем создаем сервис для запуска демона. Перейдём в директорию /etc/systemd/system/. В ней создаём сервим relaxTimer.service. Пример сервиса представлен на рисунке 32.

**

*Рисунок 32 – Сервис*

На рисунке 32 представлено описание сервиса. Description – краткое описание сервиса. [Service] – секция с параметрами запуска службы. Type – тип службы (oneshot – разовый запуск программы). ExecStart – путь до исполняемого файла. StandardOutput – управление выводом стандартного потока (tty – вывод осуществляется в терминал). TTYReset – сброс терминальных настроек перед запуском сервиса. User – пользователь, от чьего имени будет запущен юнит.

Для периодичного исполнения файла создадим таймер. Таймеры создаются в той же директории, что и сервис. Пример созданного таймера Timer.timer представлен на рисунке 33.

**

*Рисунок 33 – Таймер*

На рисунке 33 представлено описание таймера. OnUnitActiveSec – отсчет относительно момента запуска юнита (20 минут). Unit – ссылка на сервис, который будет запущен таймером. AccuracySec – точность таймера. WantedBy – определяет таргет, при активации которого будет запущен таймер.

Для перезагрузки демонов используем команду systemctl daemon-reload. Затем с помощью команды systemctl start запускаем сначала таймер, а потом сервис. Результат работы демона представлен на рисунке 34.

**

*Рисунок 34 – Работа демона*

# **Вывод**

В ходе выполнения работы был приобретён опыт работы с файлами и каталогами в ОС Linux, настройки прав на доступ к файлам и каталогам.

**Контрольные вопросы**

1. Что такое файловая система?

Файловая система — это способ организации и хранения данных на носителе информации, таком как жесткий диск, SSD или флеш-накопитель. Она определяет, как данные будут записываться, считываться и организовываться, а также управляет доступом к ним.2. Права доступа к файлам.

Каждому файлу и каталогу в системе может быть присвоено три типа владельцев, а именно пользователь, группа, другой (все).

Владелец-пользователь – это, как правило, тот, кто создал файл или каталог.

Владелец-группа – это пользователи, обладающие равными правами на доступ к конкретному файлу.

Владелец-Другой – это тот пользователь, который не создавал конкретный файл и не входит в группу, которой разрешено выполнять с файлом какие-либо действия.

Для изменения прав доступа к файлам/каталогам используется команда chmod (change mode), её общий формат: chmod <права\_доступа> <имя\_файла>. У данной команды два режима работы: числовой и символьный. В числовом режиме права доступа кодируются трехзначным числом, каждая из цифр которого от 0 до 7, последовательно для пользователя, группы, всех остальных.

3. Что такое символическая ссылка?

Символическая ссылка – это специальный файл в файловой системе, в котором вместо пользовательских данных содержится путь к файлу, открываемому при обращении к данной ссылке.

4. Что такое жесткая ссылка?

Жесткая ссылка является своего рода синонимом для существующего файла. Когда создается жесткая ссылка, то создается дополнительный указатель на существующий файл, но не копия файла. Жесткие ссылки выглядят в файловой структуре как еще один файл.

5. Команда поиска в Linux. Основные сведения.

Команда find используется поиска файлов и каталогов на основе специальных условий. Ее можно использовать в различных обстоятельствах, например, для поиска файлов по разрешениям, владельцам, группам, типу, размеру и другим подобным критериям. Общий формат: find <каталог> <опции> <критерий> <шаблон>.

6. Перечислите основные команды работы с каталогами.

pwd – отображение текущего каталога;

cd – перемещение по каталогам;

mkdir – создание нового каталога;

rmdir – удаление каталога.

7. Чем отличается вывод команд ls -F и ls -la?

ls -ls – отображает в подробном формате (будет отображаться владелец, группа, дата создания, размер и другие параметры) все файлы, включая скрытые.

ls -F – показывает тип объекта, к каждому объекты будет добавлен один из специализированных символов "\*/=>@|".

8. С помощью какой команды можно переместить файл в другой каталог?

Команда mv перемещает файл в новое место. Она также может использоваться для переименования файлов. Общий формат: mv <источник> <назначение>.

9. Куда вы переходите, выполнив команду cd без параметров?

При выполнении команды cd без параметров рабочей папкой будет выбран домашний каталог.

10. Как осуществить просмотр каталогов и их содержимого?

Команда ls отобразит каталоги нужной директории, при задании опции -l можно получить подробную информацию о файлах/каталогах, а опция -R позволит посмотреть каталог рекурсивно, то есть со всеми содержащимися в нем каталогами.

11. Как осуществить создание нового каталога и необходимых каталогов рекурсивно?

Для рекурсивного создания каталога используется команда mkdir с параметром -p (создаст родительские каталоги при их отсутствии).

12. Как осуществить рекурсивное копирование всех файлов из одного каталога в другой?

Для рекурсивного копирования одного файлов одного каталога в другой необходимо использовать команду cp с опцией -r.

13. Как рекурсивно удалить все файлы и подкаталоги в определенном каталоге?

Для рекурсивного удаления файлов и каталогов определенного каталога необходимо использовать команду rm с параметром -r.

14. Перечислите основные ключи команды ls с их назначением.

Команда ls в Unix-подобных системах используется для отображения списка файлов и каталогов. Вот основные ключи и их назначения:

1. -a: Показывает все файлы, включая скрытые (начинающиеся с точки).

2. -l: Выводит детализированный список файлов с информацией о разрешениях, владельце, размере и дате изменения.

3. -h: Используется вместе с -l для отображения размеров файлов в удобочитаемом формате (например, KB, MB).

4. -R: Рекурсивно перечисляет файлы и каталоги во всех подкаталогах.

5. -t: Сортирует файлы по времени последнего изменения, показывая самые новые файлы первыми.

6. -S: Сортирует файлы по размеру, показывая самые большие файлы первыми.

7. -r: Обратный порядок сортировки (например, от больших к маленьким или от самых новых к самым старым).

8. -d: Показывает информацию только о каталоге, а не о его содержимом.

9. -1: Выводит каждый файл на отдельной строке.

15. Команды tee и cat. Назначение и применение. Чем cat отличается от more и less?

Команда tee – считывает данные из стандартного ввода и записывает их одновременно в файл и на стандартный вывод. Используется для создания копий вывода команд, например, при записи вывода в файл, не прерывая отображение в терминале.

Команда cat – выводит содержимое файла на экран или соединяет несколько файлов в один. Показывает все содержимое файла сразу, без возможности постраничного просмотра.

Команда more – показывает содержимое файла постранично, то есть присутствует возможность пролистывания выводимого содержимого (но только вниз).

Команда less – более мощная версия, по сравнению с more, так как позволяет листать содержимое как вперед, так и назад и работает быстрее для больших файлов, так как загружает их частями.

16. Что такое демон?

Демон – это процесс, который работает в фоновом режиме без прямого участия пользователя.

17. Для чего в операционной системе Linux применяется подсистема systemD?

Система инициализации (systemD) – это система в Linux, которая подготавливает к работе операционную система. Система инициализации запускается ядром как первый процесс операционной системы. Затем, этот первый процесс, запускает все остальные процессы. Также, при выключении компьютера, данная система инициализации занимается остановкой всех процессов.

18. Какие типы юнитов вы знаете?

Все задачи, которые выполняет systemD описываются в специальных файлах, которые и называются юниты. Юниты бывают разных типов:

- Service – описывает сервис (службу) или скрипт, то есть все, что можно запустить;

- Target – группирует юниты, с его помощью можно объединить две группы и запускать как одну;

- Timer – определяет таймер, то есть службы могут запускаться по определенному расписанию, или с задержкой.

19. С помощью каких команд осуществляется управление демонами?

Управление демонами в Unix-подобных системах обычно осуществляется с помощью следующих команд:

1. systemctl: Используется в системах с systemd для управления службами (демонами). Примеры:

• systemctl start имя\_службы: Запускает службу.

• systemctl stop имя\_службы: Останавливает службу.

• systemctl restart имя\_службы: Перезапускает службу.

• systemctl status имя\_службы: Показывает статус службы.

• systemctl enable имя\_службы: Включает автозапуск службы при загрузке.

• systemctl disable имя\_службы: Отключает автозапуск службы.

2. service: Используется в системах, где управляют службами через init.d или upstart. Примеры:

• service имя\_службы start: Запускает службу.

• service имя\_службы stop: Останавливает службу.

• service имя\_службы restart: Перезапускает службу.

• service имя\_службы status: Проверяет статус службы.

2. pkill и kill: Используются для завершения работы демонов по их именам или PID (идентификатор процесса). Примеры:

• pkill имя\_демона: Завершает все процессы с указанным именем.

• kill PID: Завершает процесс с указанным PID.

Эти команды могут варьироваться в зависимости от используемой системы и версии дистрибутива.

20. Чем корутина отличается от потоков?

Кроутины и потоки – это разные подходы к многозадачности. Основное различие заключается в следующем:

- Потоки создаются операционной системой, каждый поток выполняется независимо и может использовать несколько процессорных ядер одновременно. Потоки работают параллельно, разделяя общие ресурсы и память.

- Корутины представляют собой функции, которые могут приостанавливать выполнение (сохраняя свое состояние) и возобновлять его позже. В отличие от потоков, они управляются на уровне программы и не требуют участия операционной системы. Они реализуют кооперативную многозадачность (функция приостанавливает выполнение сама, а не по решению ОС).

21. Почему корутины должны быть реализованы как выделенная языковая возможность?

Корутины требуют выделенной поддержки на уровне языка программирования, потому что их реализация напрямую затрагивает поведение функций, управление стеками вызовов и управление состояниями программы. Операции по приостановке и возобновлению выполнения, а также сохранение контекста требуют изменений в способе, которым язык управляет функциями и памятью.

22. Какая польза от корутины?

Корутины полезны для написания асинхронного и неблокирующего кода, который выглядит и ведет себя как обычный синхронный код. Они позволяют разработчикам избегать сложностей с обратными вызовами, упрощая управление состояниями программы. Корутины также позволяют более эффективно использовать системные ресурсы, так как не требуют создания потоков и затрат на переключение контекста. Это делает их особенно полезными для задач, которые предполагают ожидание, например, сетевые запросы, когда использование потоков было бы излишним и менее эффективным.