**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Программа ДПО Intaro

ЛАБОРАТОРНАЯ РАБОТА №2

“Работа с файловой системой ОС Linux”

Студент \_\_\_\_\_\_\_\_\_\_ Целищев А.Е.

ПМ-21-2

Руководитель \_\_\_\_\_\_\_\_\_\_ Кургасов В.В.

Доцент каф. АСУ

Липецк 2023 г

Оглавление

[Цель работы 3](#_Toc9114)

1. [Общая часть](#_Toc12389) 4

[2. Файлы и каталоги 11](#_Toc27089)

[3. Пользователи и группы 1](#_Toc4632)8

[4. Архивация и поиск 2](#_Toc32722)1

[Выводы 2](#_Toc14593)2

[Ответы на контрольные вопросы 2](#_Toc19725)3

**Цель работы**

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

1. **Общая часть**

1.2. FHS (Filesystem Hierarchy Standard) - стандарт, унифицирующий местонахождение файлов и каталогов с общим назначением в файловой системе UNIX. Большинство UNIX-подобных систем так или иначе следует этому стандарту, хотя есть Linux-дистрибутивы, отвергающие этот стандарт, например, GoboLinux. В то же время, например, Mac OS X, использует собственные имена /Library/, /Applications/, /Users/ вместе с традиционные именами UNIX-иерархии.

В FHS все файлы и каталоги находятся внутри корневого каталога, даже если они расположены на различных физических носителях.

1.3. Перечень основных каталогов согласно FHS

|  |  |
| --- | --- |
| Путь | Описание |
| / | Корневой каталог, содержащий всю файловую иерархию. |
| /bin | Основные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям. |
| /boot | Загрузочные файлы (файлы загрузчика, ядро, initrd, System.map) |
| /dev | Основные файлы устройств |
| /etc | Общесистемные конфигурационные файлы |
| /etc/opt | Файлы конфигурации для opt |
| /etc/X11 | Файлы конфигурации X Windows System версии 11 |
| /etc/sgml | Файлы конфигурации SGML |
| /etc/xml | Файлы конфигурации XML |
| /home | Содержит домашние каталоги пользователей, которые в свою очередь содержат персональные настройки и данные пользователя. |
| /lib | Основные библиотеки, необходимые для работы программ из /bin и /sbin |
| /media | Точки монтирования для сменных носителей (CD-ROM, DVD-ROM и тд) |
| /mnt | Содержит временно монтируемые файловые системы |
| /opt | Дополнительное программное обеспечение |
| /proc | Виртуальная файловая система, представляющая состояние ядра операционной системы и запущенных процессов в виде файлов |
| /root | Домашний каталог пользователя root |
| /run | Информация о системе с момента ее загрузки, в том числе данные, необходимые для работы демонов (pid-файлы, UNIX-сокеты и тд.) |
| /sbin | Основные системные программы для администрирования и настройки системы |
| /srv | Данные для сервисов, предоставляемых системой (например, www или ftp) |
| /sys | Содержит информацию об устройствах, драйверах, а также некоторых свойствах ядра |
| /tmp | Временные файлы |
| /usr | Вторичная иерархия для данных пользователя. Содержит большинство пользовательских приложений и утилит, используемых в многопользовательском режиме. Может быть смонтирована по сети только для чтения и быть общей для нескольких машин |
| /usr/bin | Дополнительные программы для всех пользователей, не являющиеся необходимыми в однопользовательском режиме |
| /usr/include | Стандартные заголовочные файлы |
| /usr/lib | Библиотеки для программ, находящихся в usr/bin и usr/sbin |
| /usr/local | Третичная иерархия для данных, специфичных для данного хоста. Обычно содержит такие подкаталоги, как bin, lib, share |
| /usr/sbin | Дополнительные системные программы (например, демоны сетевых сервисов) |
| /usr/share | Архитектурно-независимые общие данные |
| /usr/src | Исходные коды (например, здесь располагаются исходные коды ядра) |
| /var | Изменяемые файлы, такие как файлы регистрации, временные почтовые файлы и тд |
| /var/cache | Кэш приложений. Такие данные генерируются локально в результате ресурсозатратных вычислений или операций ввода-вывода. Приложение обязано уметь регенерировать эти данные. Данные файлы могут быть удалены без потери данных |
| /var/lib | Информация о состоянии. Постоянные данные, изменяемые программами в процессе работы (например, базы данных, метаданные пакетного менеджера и др.) |
| /var/lock | Файлы блокировки, указывающие на занятость некоторого ресурса |
| /var/log | Различные файлы регистрации |
| /var/mail | Почтовые ящики пользователей |
| /var/run | Каталог для хранения информации о системе с момента её загрузки, на данный момент устарел. Допускается делать его ссылкой на /run. Каталог оставлен для обратной совместимости с программами, которые всё ещё используют каталог /var/run |
| /var/spool | Задачи, ожидающие обработки (например, очереди печати, непрочитанные или неотправленные письма) |
| /var/spool/mail | Местоположение пользовательских почтовых ящиков (устаревшее) |
| /var/tmp | Временные файлы, которые должны быть сохранены между перезагрузками |

1.4. Зайдем в терминал под root, введя sudo bash и пароль.



Рис 1. Вход в терминал под root.

1.5. Введем lsblk чтобы вывести содержимое каталога файлов физических устройств:

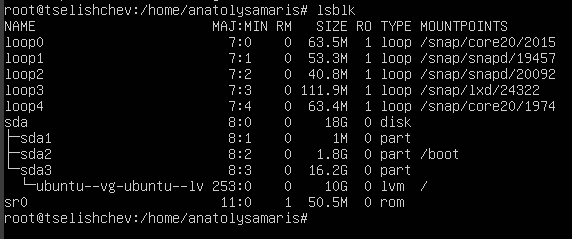


Рис 2. Вывод содержимого каталога файлов физических устройств.

В списке устройств видим имена loopX, sda, sr0.

- sda: это обозначение для первого блочного устройства, обычно это жесткий диск или твердотельный накопитель. "sdb" будет вторым устройством, "sdc" - третьим и так далее.

- loop: это обозначение для устройств loopback, которые позволяют монтировать файлы и образы как блочные устройства. Это может использоваться, например, для монтирования образов дисков или файловых систем в файле.

- sr0: это обозначение для устройств CD/DVD-ROM. "sr0" обычно является первым устройством CD/DVD-ROM в системе. Если есть несколько устройств CD/DVD-ROM, то они могут быть обозначены как "sr1", "sr2" и так далее.

1.6. Перейдем в директорию пользователя root, просмотрим содержимое. Нас интересует местонахождение файла vmlinuz, который должен быть в каталоге boot. Убедимся в этом:

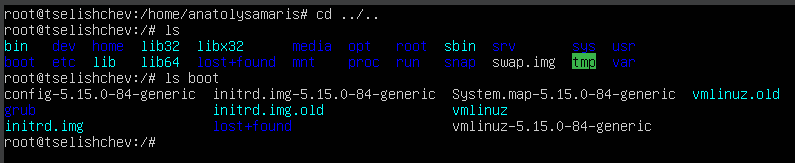


Рис 3. Переход в директорию пользователя и проверка директории boot.

Откроем файл vmlinuz, используя текстовый редактор Vim.

Командой ls -l vmlinuz просмотрим права доступа к файлу. Видим вывод lrwxrwxrwx. Здесь последовательно выводятся права доступа сначала для владельца файла, потом для группы пользователей, привязанных к файлу, и затем для всех остальных пользователей. Символ «l» перед правами доступа означает тип файла. Утилита ls обозначает символом «l» символические ссылки. Таким образом, каждый имеет доступ на чтение (“r”), запись (“w”) и исполнение (“x”) символической ссылки vmlinuz.



Рис 4. Просмотр прав доступа к файлу.

1.7. Создадим нового пользователя user командой adduser user

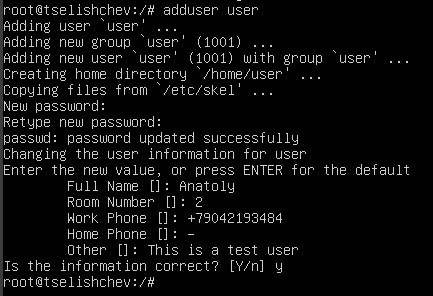


Рис 5. Создание пользователя user.

1.8. Перейдем в профиль пользователя user и создадим 3 текстовых файла, используя touch, cat, и vim

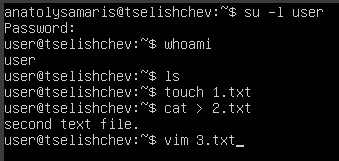


Рис 6. Вход в профиль user и создание текстовых файлов различными способами.

Используя команду ls -l мы можем посмотреть тип файла и права доступа к файлу. Как можем видеть, все созданные файлы имеют -rw-rw-r--. «-» в начале означает, что это обычный файл (а расширение .txt могло бы подсказывать системе, какой именно утилитой/программой обрабатывать файл), дальше следуют разрешения на чтение и запись для владельца (создателя) файла и группы пользователей, к которой принадлежит владелец (в нашем случае группа пользователей, в которую входит user, состоит только из него), и разрешение лишь на чтение для всех остальных пользователей.

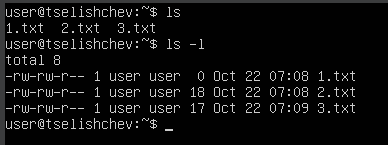


Рис 7. Просмотр типа созданных файлов прав доступа к ним.

1.9. Перейдем к директории пользователя root. Видим то же, что в пункте 1.6.



Рис 8. Директория пользователя root.

1.10. Изменим права доступа к файлу 1.txt командой chmod ugo+rwx 1.txt, которая выдаст все права всем пользователям (u - владелец, g - группа владельца, o - другие пользователи; + - добавить права, r - чтение, w - запись, x - исполнение)

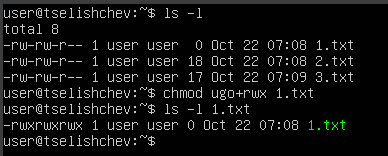


Рис 9. Изменение прав доступа к файлу 1.txt.

1.11. Создадим символическую и жесткую ссылки на файл 2.txt командами

ln -s 2.txt <имя\_ссылки> и ln 2.txt <имя ссылки> соответственно.

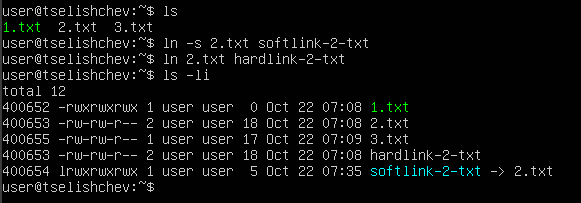


Рис 10. Создание символической и жесткой ссылок на файл 2.txt.

1.12. Создадим каталог new командой mkdir <имя\_каталога>

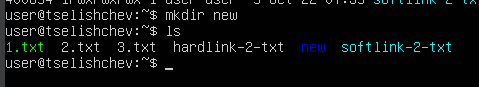


Рис 11. Создание каталога new.

1.13. Копируем файл 1.txt в каталог new с новым именем copy-1.txt с помощью команды cp 1.txt new/copy-1.txt



Рис 12. Копирование файла 1.txt в каталог new с новым именем copy-1.txt.

1.14. Переместим файл 2.txt в каталог new командой mv 2.txt new/



Рис 13. Перемещение файла 2.txt в каталог new.

1.15. Зайдем в профиль root-пользователя и сделаем его владельцем файла 3.txt и каталога new chown <имя\_пользователя> <имя\_файла\_или\_каталога>

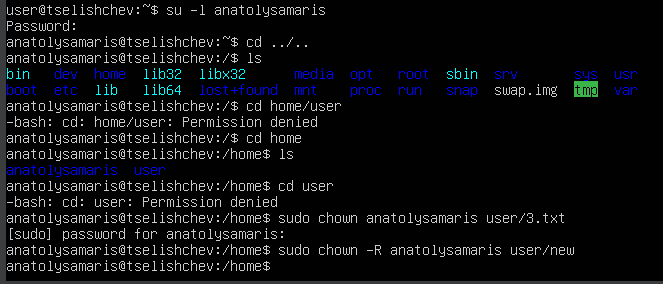


Рис 14. Изменение владельца файла 3.txt и каталога new.

1.16-1.17. Удалим копию файла 1.txt из каталога new, а затем и сам каталог new командами rm <имя\_файла> и rm -Rf <имя\_каталога>. Здесь для удаления каталога используем rm -Rf вместо rmdir чтобы рекурсивно (-R) пройти по всем имеющимся в каталоге файлам и удалить их, не спрашивая разрешения (-f). Затем будет удален и сам каталог.

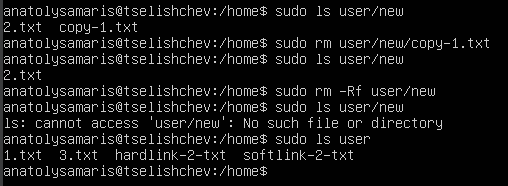


Рис 15. Удаление файла copy-1.txt и каталога new.

1. **Файлы и каталоги**
   1. Создадим 3 текстовых файла разными способами. Для создания с помощью редактора vim используем команду vi <имя\_файла>0

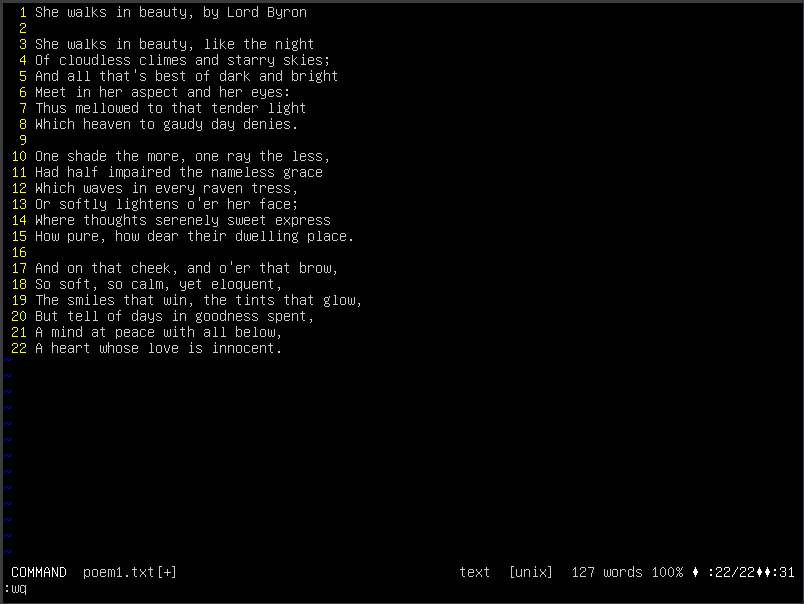


Рис 16. Создание текстового файла с помощью vim.

Прежде чем использовать mc (Midnight Commander), установим соответствующую программу: sudo apt install mc

Затем, используя комбинацию клавиш Shift + F2 (в некоторых источниках F4, хотя здесь это иная функция), создадим новый файл, выберем текстовый редактор mc для редактирования файла и напишем текст, сохранив затем клавишей F2.

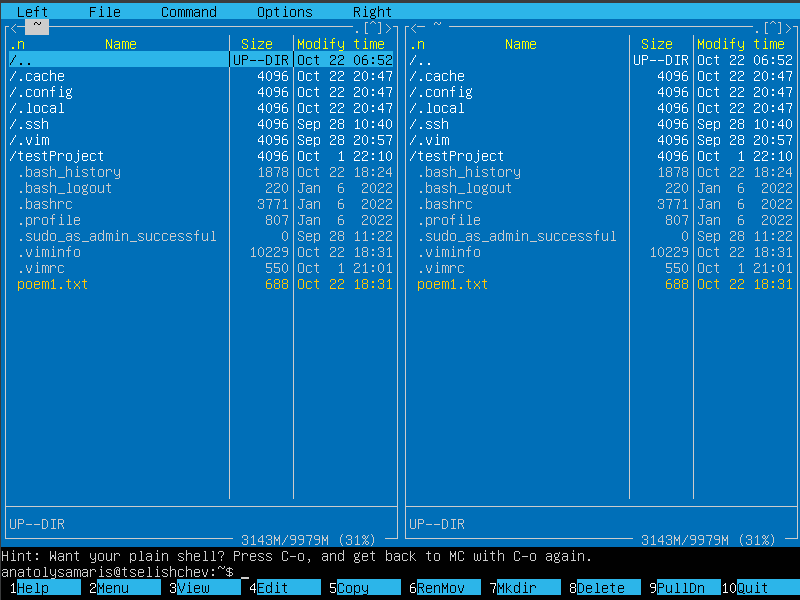


Рис 17. Интерфейс mc-редактора.

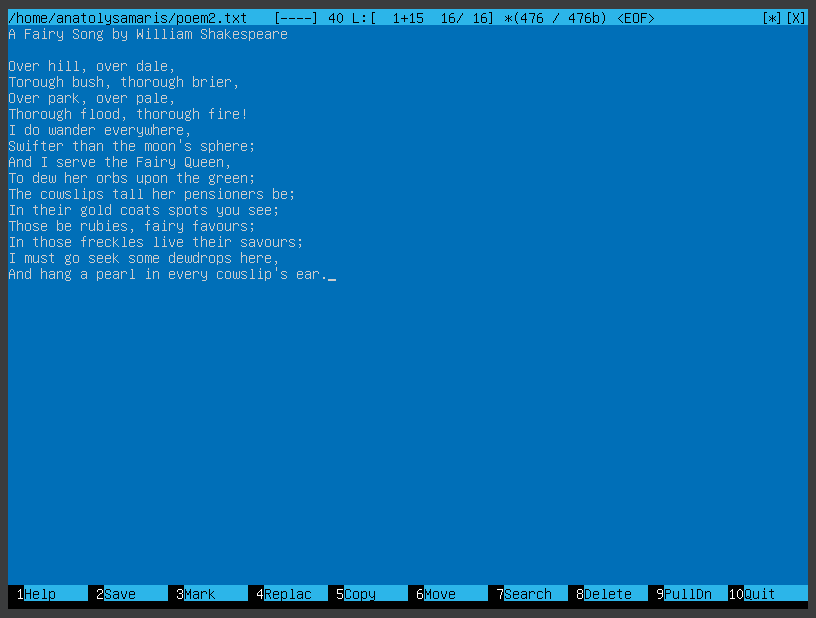


Рис 18. Создание текстового файла с помощью mc.

Создадим файл, используя tee. В целом tee используется для вывода результата команд в файлы (либо используется в комбинации с другими командами). Поэтому при записи текста в файл с помощью tee мы будем наблюдать, что каждая введённая строка тут же выводится в терминал - это особенность работы tee при простейшей записи в файл.

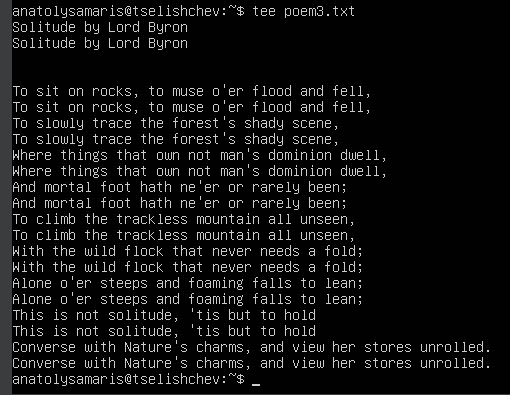
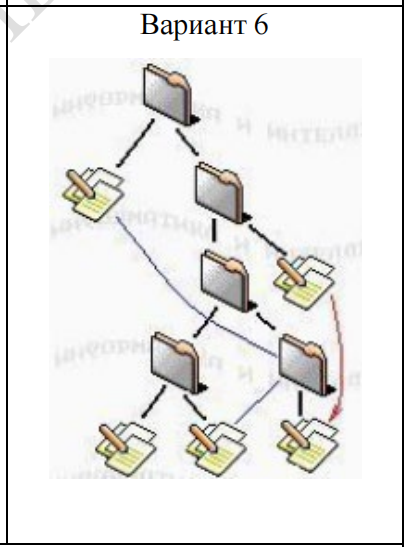


Рис 19. Создание текстового файла с помощью tee.

2.2. Создадим структуру каталогов согласно варианту (вариант 6).



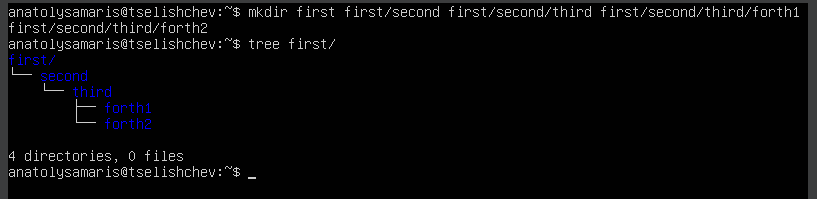


Рис 20. Создание каталогов.

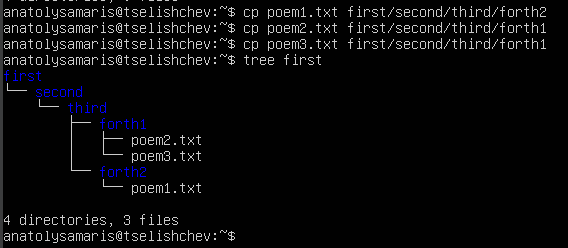


Рис 21. Создание файлов посредством копирования ранее созданных.

2.3.

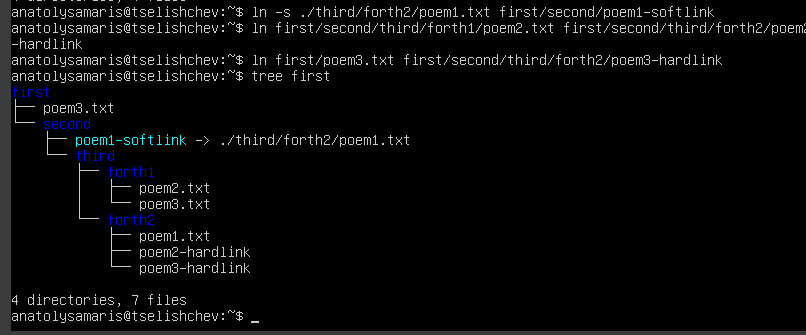


Рис 22. Создание ссылок.

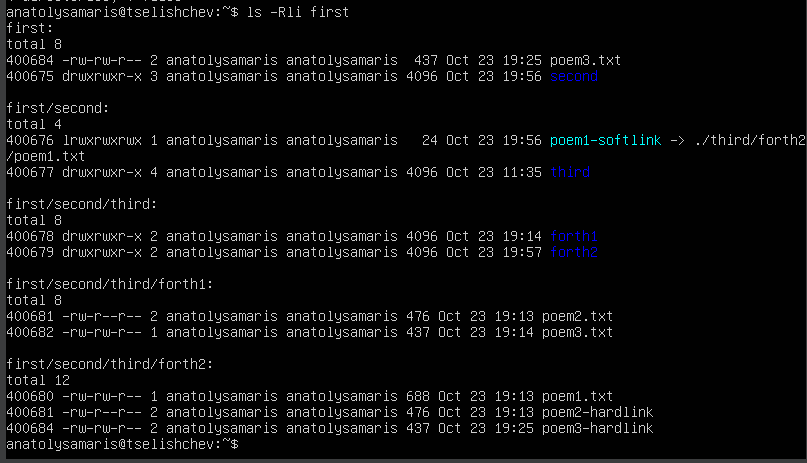


Рис 23. Демонстрация результата.

2.4.

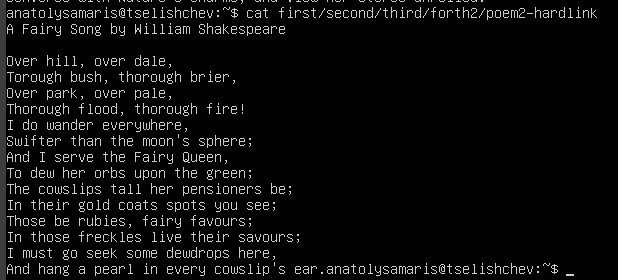


Рис 24. Демонстрация работы жесткой ссылки.

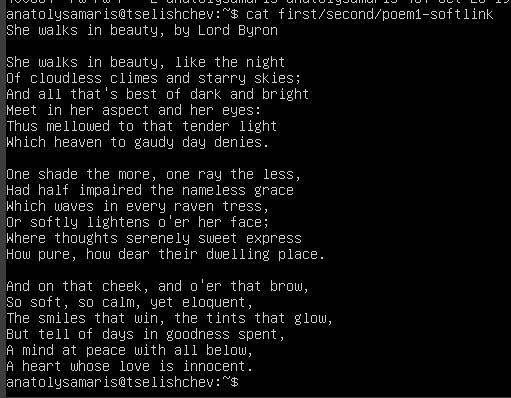


Рис 25. Демонстрация работы символической ссылки.

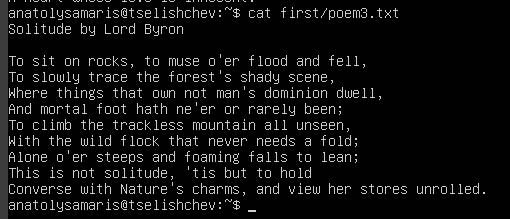


Рис 26. Обычное чтение содержимого файла.

2.5.

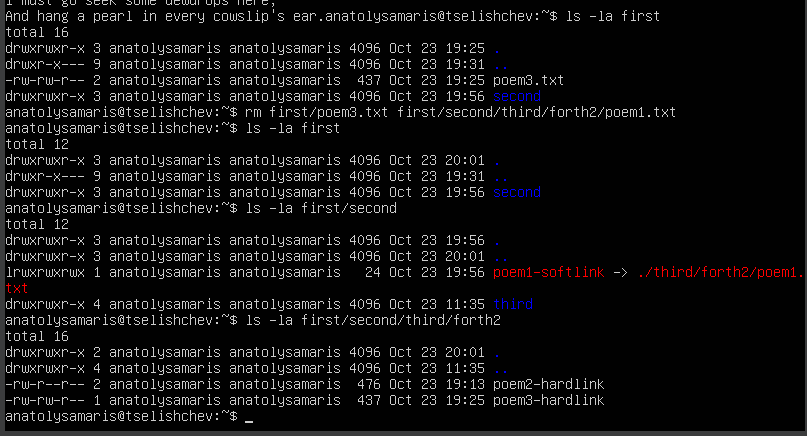


Рис 27. Демонстрация результатов удаления файлов, на которые ссылаются ссылки.

2.6.

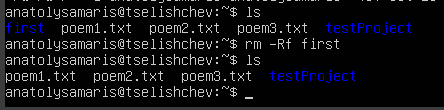


Рис 28. Рекурсивное удаление подкаталогов в first и последующее удаление этого каталога.

1. **Пользователи и группы**

3.1. Создадим пользователей anatolypm21 (МоёИмяГруппа) и tselishchevevgeny (ФамилияИмяОтца)

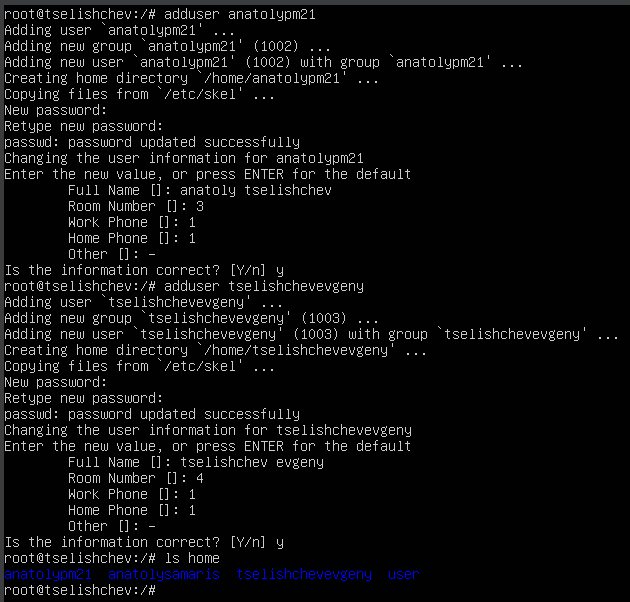


Рис 29. Создание пользователей anatolypm21 и tselishchevevgeny.

3.2. Зайдем в профиль anatolypm21, создадим файл anatolypm212, впишем в него echo «This is test page Tselishchev Anatoly Evgenevich» и сохраним файл.



Рис 30. Запись в файл anatolypm212.txt.

3.3. При попытке переместить созданный файл в каталог пользователя tselishchevevgeny мы получим ошибку - пользователь не имеет sudo-доступа, который требуется для таких манипуляций. Поэтому зайдем под root и выдадим новым пользователям возможность использовать sudo командой usermod -aG sudo <ИмяПользователя> и лишь после этого сможем переместить файл anatolypm212 из каталога пользователя anatolypm21 в каталог пользователя tselishchevevgeny



Рис 31. Выдача sudo-прав новым пользователям.

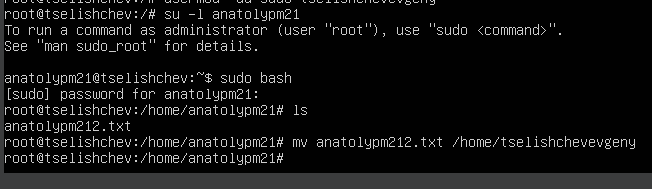


Рис 32. Перемещение файла в домашний каталог другого пользователя.

3.4. Зайдем в профиль tselishchevevgeny, обнаружим появившийся в его домашнем каталоге файл anatolypm212 и впишем в конец следующую строку: echo «Test page edited by user Tselishchev Evgeny» а также добавим в начало #!/bin/bash. Переместим файл обратно к пользователю anatolypm21.



Рис 33. Редактирование перемещенного файла.

3.5. Зайдем в профиль anatolypm21 и исполним редактированый файл anatolypm212 с помощью команды sh anatolypm212.txt

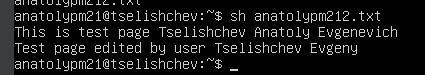


Рис 34. Исполнение итогового файла anatolypm212.txt.

1. **Архивация и поиск**

Выполним задание по варианту (вариант 6)

|  |  |
| --- | --- |
| Вариант | Задание |
| 6 | Переупаковать архив arch6.tar.bz2, содержащий несколько файлов, в архив arch6.tar.bz2 с максимальной степенью сжатия, без разархивации файлов на диск.  Вывести на экран все обычные (регулярные) файлы текущего каталога (без обработки подкаталогов). |

Сначала создадим начальный архив, собрав в него ранее созданные текстовые файлы, используя утилиту tar:

tar -cvf arch6.tar.bz2 poem1.txt poem2.txt poem3.txt

Где -c говорит о создании нового архива, -v указывает показать процесс архивации (увидим архивируемые файлы/каталоги), -f позволяет указать имя архива (либо путь + имя).

Чтобы переупаковать архив с максимальной степенью сжатия, будем использовать утилиту для сжатия bzip2 (расширение нашего архива явно говорит о том, что использовать следует его), который работает медленнее своего аналога gzip2, но производит максимальное сжатие архива, что нам и требуется. Используем команду tar -cfj arch6.tar.bz2 где флаг -j говорит об использовании bzip2.

Убедимся, что полученный архив содержит наши текстовые файлы, используя tar с флагом -t, который показывает содержимое архива:

tar -tf arch6.tar.bz2

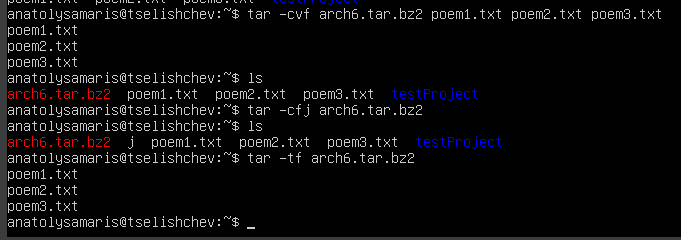


Рис 35. Действия по архивации файлов по варианту.

**Выводы**

Изучил стандарт FHS, выполнил задания по вариантам, связанные с манипуляциями с файлами и каталогами и правами доступа к ним, созданием и управлением пользователями, архивацией файлов и каталогов.

**Ответы на контрольные вопросы**

**Что такое файловая система?**

Файловая система - порядок, определяющий способ организации, хранения и именования данных на [носителях информации](https://ru.wikipedia.org/wiki/%D0%9D%D0%BE%D1%81%D0%B8%D1%82%D0%B5%D0%BB%D1%8C_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%B8" \o "Носитель информации).

С точки зрения [операционной системы](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0" \o "Операционная система), весь диск представляет собой набор [кластеров](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%82%D0%B5%D1%80_(%D0%B5%D0%B4%D0%B8%D0%BD%D0%B8%D1%86%D0%B0_%D1%85%D1%80%D0%B0%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)" \o "Кластер (единица хранения данных)) (как правило, размером, кратным 512 [байт](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B9%D1%82" \o "Байт)). Драйверы файловой системы организуют кластеры в файлы и каталоги (реально являющиеся файлами, содержащими список файлов в этом каталоге). Эти же драйверы отслеживают, какие из кластеров в настоящее время используются, какие свободны, какие помечены как неисправные.

**Права доступа к файлам. Назначение прав доступа.**

Права доступа являются одной из функций для обеспечения безопасности системы: во-первых, позволяют сохранять конфиденциальную информацию, ограничивая доступ для других пользователей, во-вторых, не позволяет вирусам принести вред системе, так как они не смогут получить доступ к важным файлам, и, следовательно, повредить их или использовать информацию из них во вред пользователю.

Каждый файл имеет три параметра доступа: чтение («r»), запись («w») и выполнение («x»). Помимо этого, файл имеет три категории пользователей, выделяя для каждой отдельный набор параметров доступа: владелец файла, группа пользователей (обычно это группа владельца), остальные пользователи.

Таким образом, для каждой категории пользователей мы можем настроить отдельные права доступа к файлу.

**Что такое «символическая ссылка»?**

Символические ссылки более всего похожи на обычные ярлыки. Они содержат адрес нужного файла в вашей файловой системе. Когда вы пытаетесь открыть такую ссылку, то открывается целевой файл или папка. Главное ее отличие от жестких ссылок в том, что при удалении целевого файла ссылка останется, но она будет указывать в никуда, поскольку файла на самом деле больше нет. То есть при попытке открыть такую ссылку мы получим ошибку, в которой будет сказано, что такой файл не найден (как если бы напрямую попытались работать с этим файлом).

**Что такое «жесткая ссылка»?**

Этот тип ссылок реализован на более низком уровне файловой системы. Файл размещен только в определенном месте жесткого диска. Но на это место могут ссылаться несколько ссылок из файловой системы. Каждая из ссылок - это отдельный файл, но ведут они к одному участку жесткого диска. Файл можно перемещать между каталогами, и все ссылки останутся рабочими, поскольку для них неважно имя.

**Команда поиска в Linux. Основные сведения.**

Команда find - это одна из наиболее важных и часто используемых утилит системы Linux. Это команда для 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.

Утилита find предустановлена по умолчанию во всех Linux дистрибутивах.



**Перечислите основные команды работы с каталогами.**

- ls - отображение содержимого каталога.

- cd - смена текущего каталога.

- mkdir - создание нового каталога.

- rmdir - удаление пустого каталога.

- cp - копирование файлов и каталогов.

- mv - перемещение и переименование файлов и каталогов.

- rm - удаление файлов и каталогов.

- pwd - вывод текущего каталога.

- find - поиск файлов и каталогов.

- tree - вывод древовидной структуры каталогов.

**Чем отличается вывод команд ls -F и ls -la?**

ls -F: При использовании флага -F будут добавлены служебные символы к именам файлов, указывающие их тип. Например, символ «/» добавляется к именам каталогов, «\*» - к исполняемым файлам и тд.

ls -la: флаг -a указывает отображать все файлы и каталоги, включая скрытые (начинающиеся с точки). Флаг -l команда выводит расширенную информацию о файлах и каталогах, включая различные атрибуты, такие как права доступа, владелец, группа, размер, время последней модификации и т. д.

**С помощью какой команды можно переместить файл в другой каталог?**

mv <ПутьКФайлу> <НовоеРасположениеФайла>

Причем при использовании этой команды мы можем тут же переименовать файл, добавив к новому расположению другое название файла. Также этой командой можно просто переименовать файл следующим образом:

mv <ИмяФайла> <НовоеИмяФайла>

**Куда вы переходите, выполнив команду cd без параметров?**

В домашний каталог пользователя.

**Как осуществить просмотр подкаталогов и их содержимого?**

ls -R

**Как осуществить создание нового каталога и необходимых подкаталогов рекурсивно?**

mkdir -p path/to/newDir

При использовании ключа -p, если подкаталоги path, to отсутствуют, то они будут созданы рекурсивно.

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

cp -R <ПутьКФайлам> <НовоеРасположениеФайлов>

**Как рекурсивно удалить все файлы и подкаталоге в определенном каталоге?**

rm -rf <ПутьКФайлам>

Ключ -r обозначает рекурсивное удаление файлов, -f указывает не запрашивать подтверждения при удалении каждого из файлов.

**Перечислите основные ключи команды ls с их назначением.**

- -a - отображает все файлы и папки, включая скрытые файлы, начинающиеся с точки.

- -l - выводит детальную информацию о файлах и папках в виде списка.

- -h - отображает размеры файлов в человекочитаемом формате (например, "1K", "2M", "3G" и т. д.).

- -R - рекурсивно выводит все файлы и папки в текущей и всех подпапках.

- -t - сортирует файлы по времени последнего изменения, начиная с самого нового.

- -S - сортирует файлы по размеру, начиная с самого большого.

- -r - выводит файлы в обратном порядке.

- -u - сортирует файлы по времени последнего доступа, а не последнего изменения.

- -d - отображает только имена указанных папок, не списки их содержимого.

- --color - предоставляет цветовую разметку для отображения файлов и папок.

- -G - отображает файлы и папки с использованием цветов, но только в виде списка.

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

Команда tee используется для чтения входных данных и их записи как в стандартный вывод, так и в один или несколько файлов. Она позволяет сохранить копию входных данных в файл без их потери для дальнейшего использования. Пример использования команды tee:

command | tee file.txt

В этом примере результат выполнения команды command будет выведен на стандартный вывод и записан в файл file.txt.

Команда cat (от concatenate) используется для чтения содержимого одного или нескольких файлов и вывода их содержимого в стандартный вывод. Она может также использоваться для создания, объединения или добавления содержимого в файлы. Пример использования команды cat:

cat file.txt

В этом примере содержимое файла file.txt будет выведено на стандартный вывод.

cat отличается от more и less тем, что она выводит содержимое файла целиком, без ограничений. more и less используются для поблочного вывода содержимого файла, что позволяет пролистывать его, постранично просматривая данные. more допускает только прокрутку вниз, тогда как less позволяет прокручивать содержимое файла в обе стороны.