РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Лабораторная работа №5 Анализ файловой системы Linux. Команды для работы с файлами и каталогами

Абдуллина Ляйсан Раисовна НПИбд-01-21

Содержание

# 1 Цель работы

Ознакомление с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобретение практических навыков по применению команд для работы с файлами и каталогами, по управлению процессами (и работами),по проверке использования диска и обслуживанию файловой системы.

# 2 Теоретическое введение

Например, в таблице ниже приведено краткое описание стандартных команд.

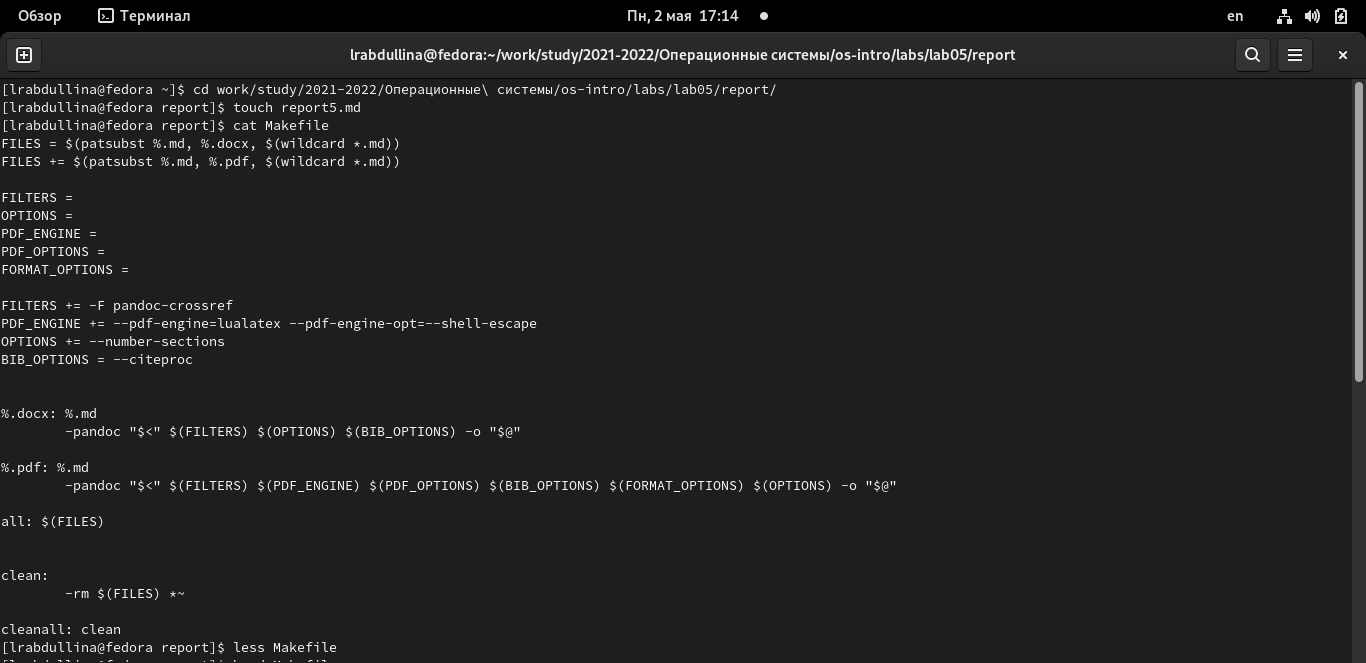
Описание некоторых основных команд.

| Основные команды | Описание команд |
| --- | --- |
| touch имя-файла | Для создания текстового файла |
| cat имя-файла | Для просмотра файлов небольшого размера |
| less имя-файла | Для просмотра файлов постранично |
| head [-n] имя-файла | Выводит по умолчанию первые 10 строк файла, где n—количество выводимых строк |
| tail [-n] имя-файла | Выводит умолчанию 10 последних строк файла, где n—количество выводимых строк |
| cp [-опции] исходный\_файл целевой\_файл | Для копирования файлов и каталогов |
| mv [-опции] старый\_файл новый\_файл | Для перемещения и переименования файлов и каталогов |
| chmod режим имя\_файла | Изменение права доступа к файлу или каталогу |

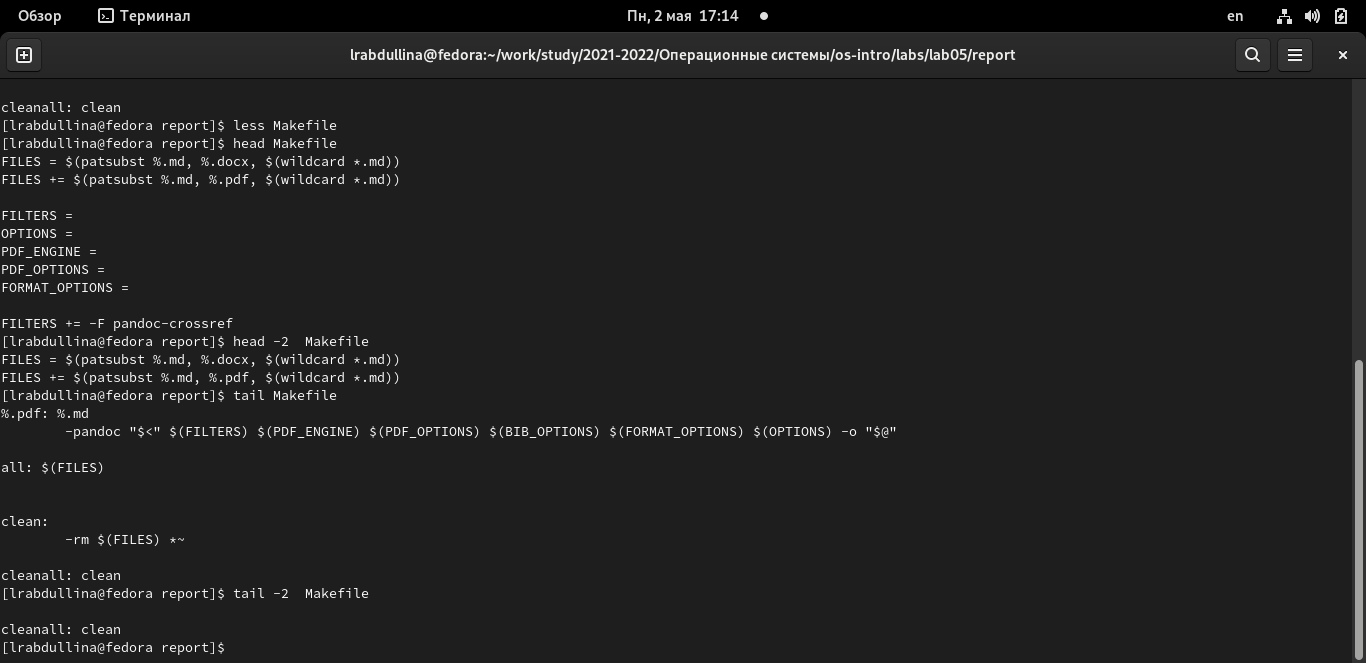
# 3 Выполнение лабораторной работы

##1

Выполним все примеры, приведённые в первой части описания лабораторной работы. Создадим файл с помощью команды touch, просмотрим содержимое файла Makefile (целиком и постранично) c помощью команд cat и less, а также выведем 10 первых и последних строчек (азатем и по 2 строчке) с помощью команд head и tails сооветсвенно. (скриншоты 1, -fig. 2])



Скриншот 1: Создание файла, просмотр файла Makefile

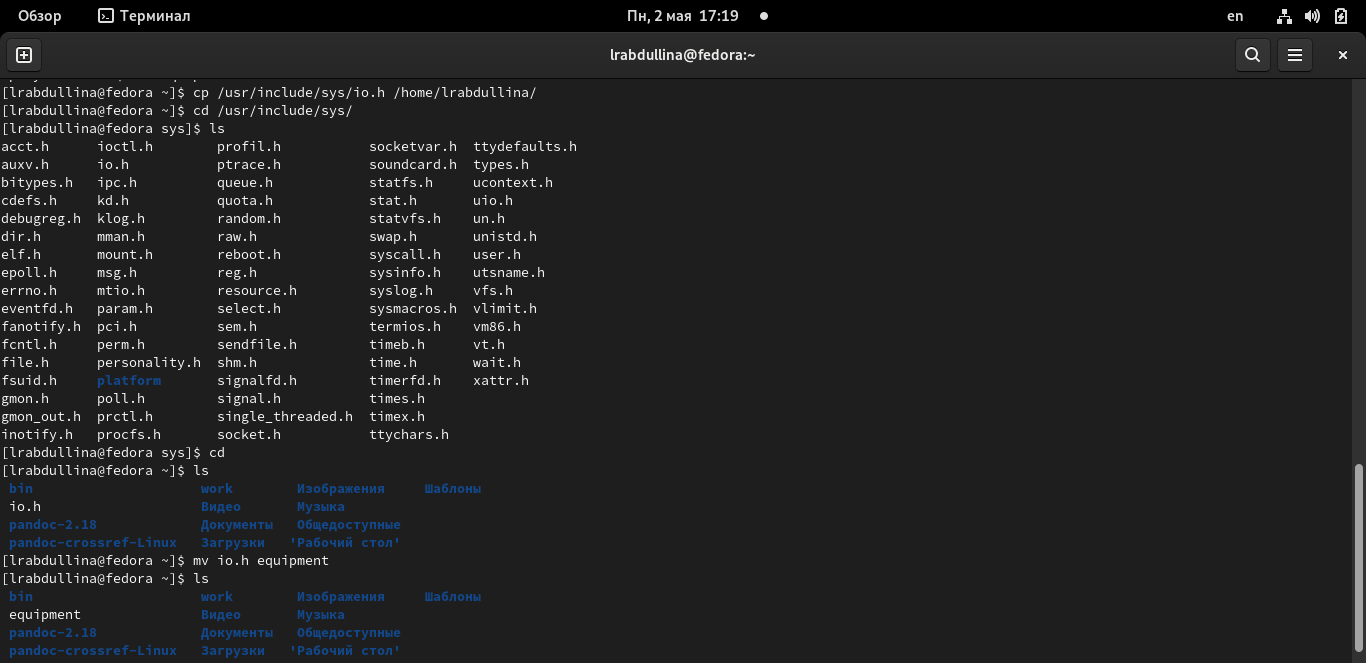


Скриншот 2: Просмотр первых и последних строчек Makefile

##2

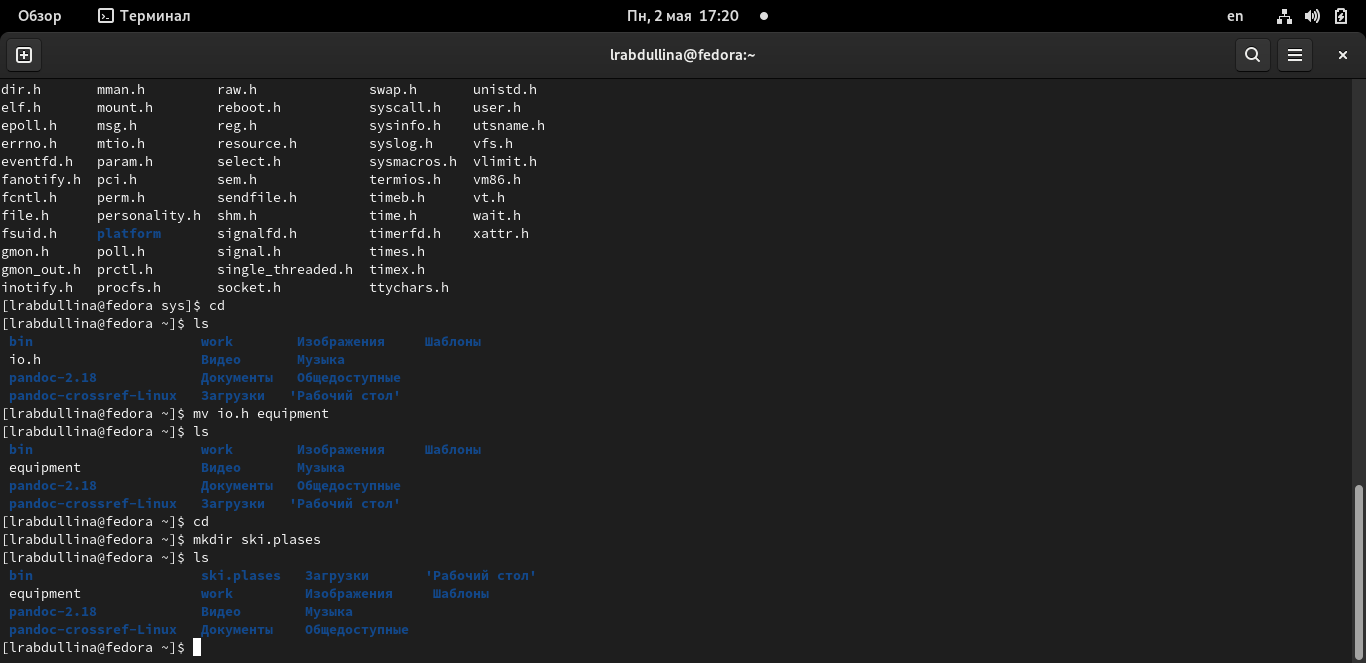
Выполним следующие действия:

1. Скопируем файл /usr/include/sys/io.h в домашний каталог с помощью команды cp и назовем его equipment с помощью команды mv. (скриншот 3)



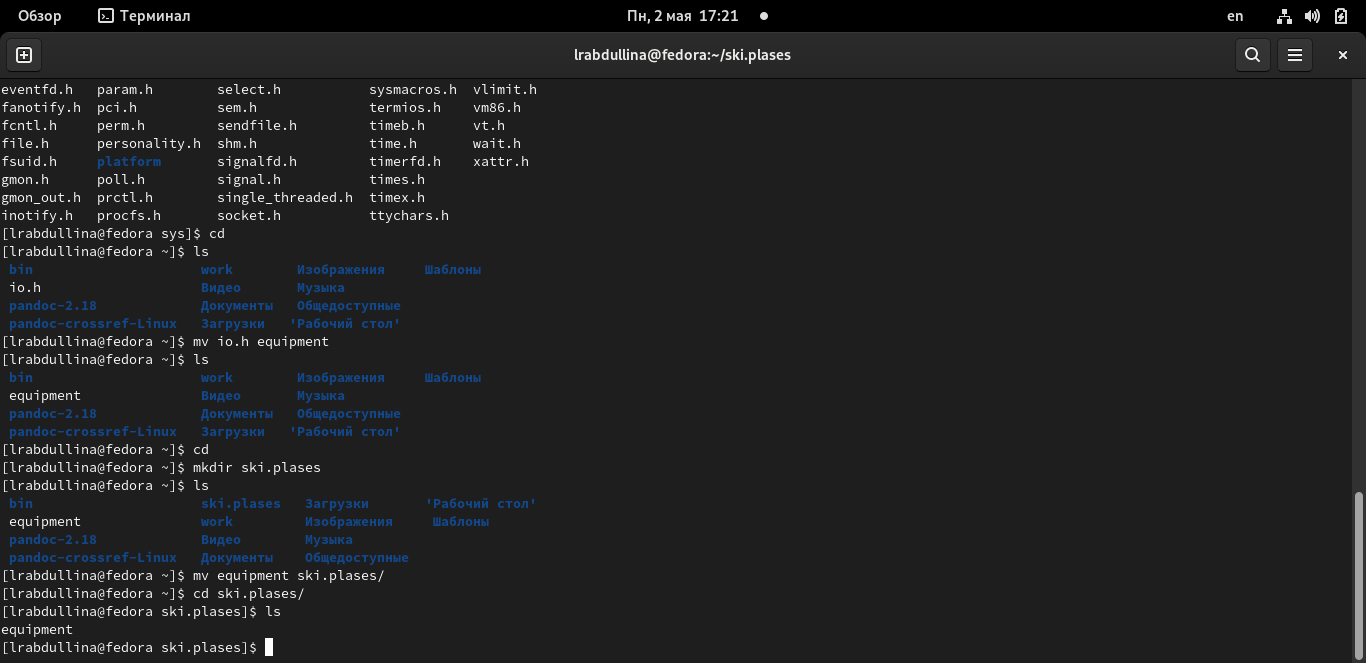
Скриншот 3: Копирование файла, проверка копирования, переименовывание файла и проверка

1. В домашнем каталоге создадим директорию ~/ski.plases с помощью команды mkdir. (скриншот 4)



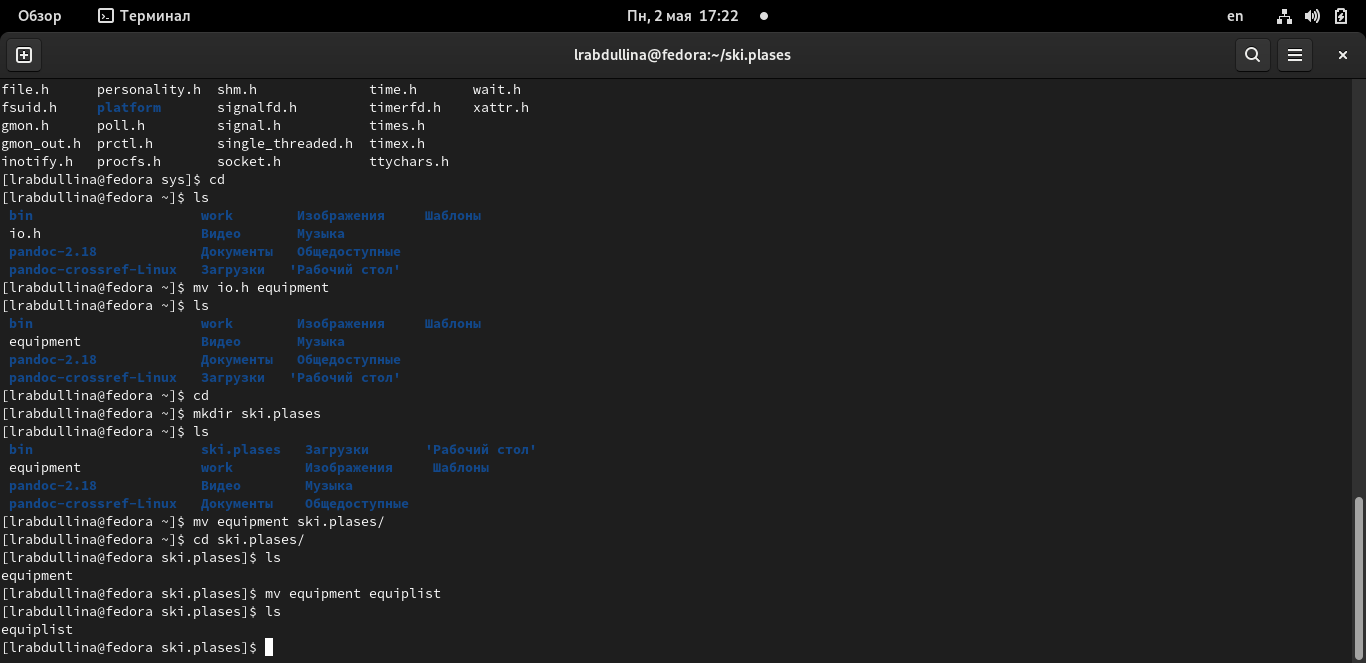
Скриншот 4: Создание директории и проверка

1. Переместим файл equipment в каталог ~/ski.plases через команду mv.(Скриншот 5)



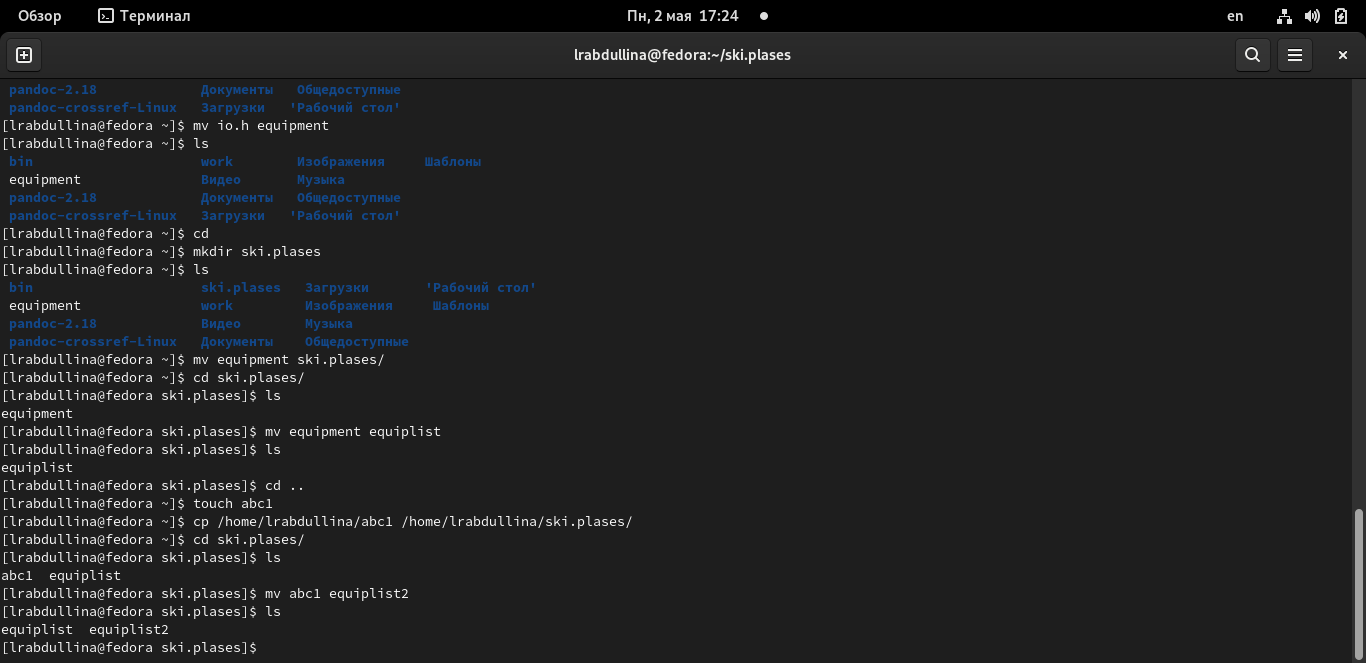
Скриншот 5: Перемещение файла, проверка

1. Переименуем файл ~/ski.plases/equipment в ~/ski.plases/equiplist черех mv. (Скриншот 6)

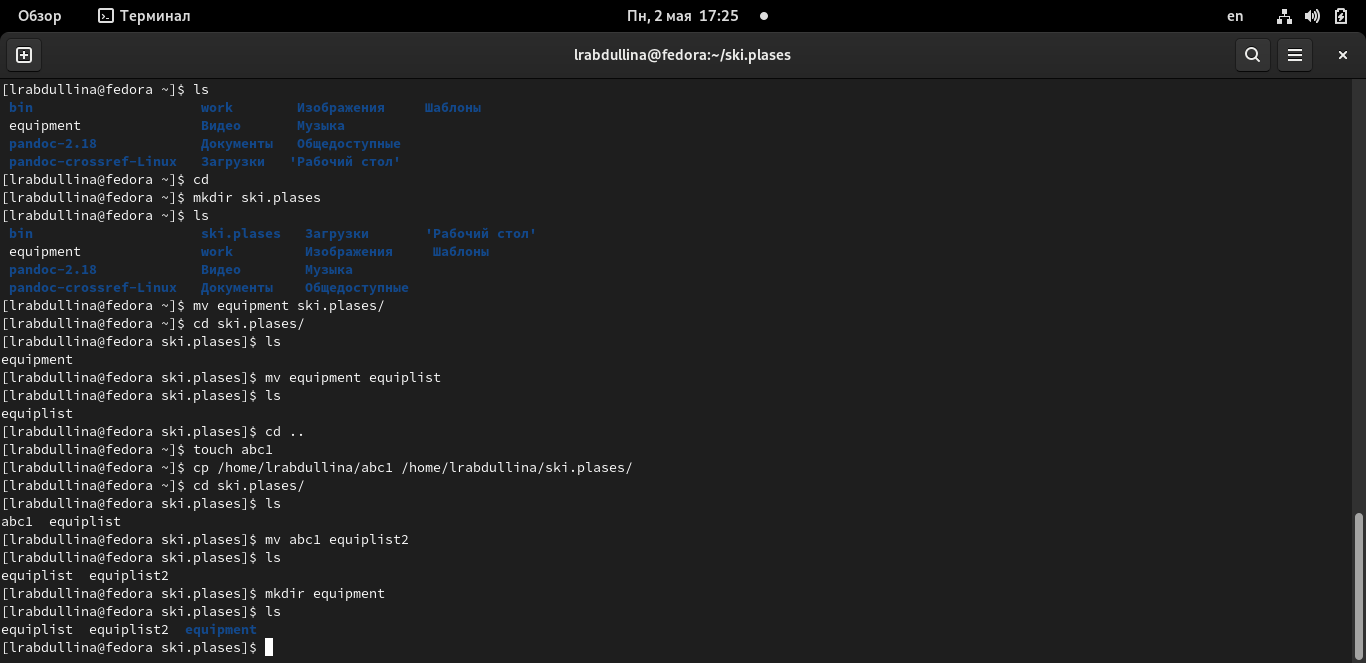


Скриншот 6: Переименовывание файла, проврека

1. Создадим в домашнем каталоге файл abc1 через команду touch и скопируем его в каталог ~/ski.plases с помощью команды cp и назовём его equiplist2 через команду mv. (Скриншот **¿fig:007?**)

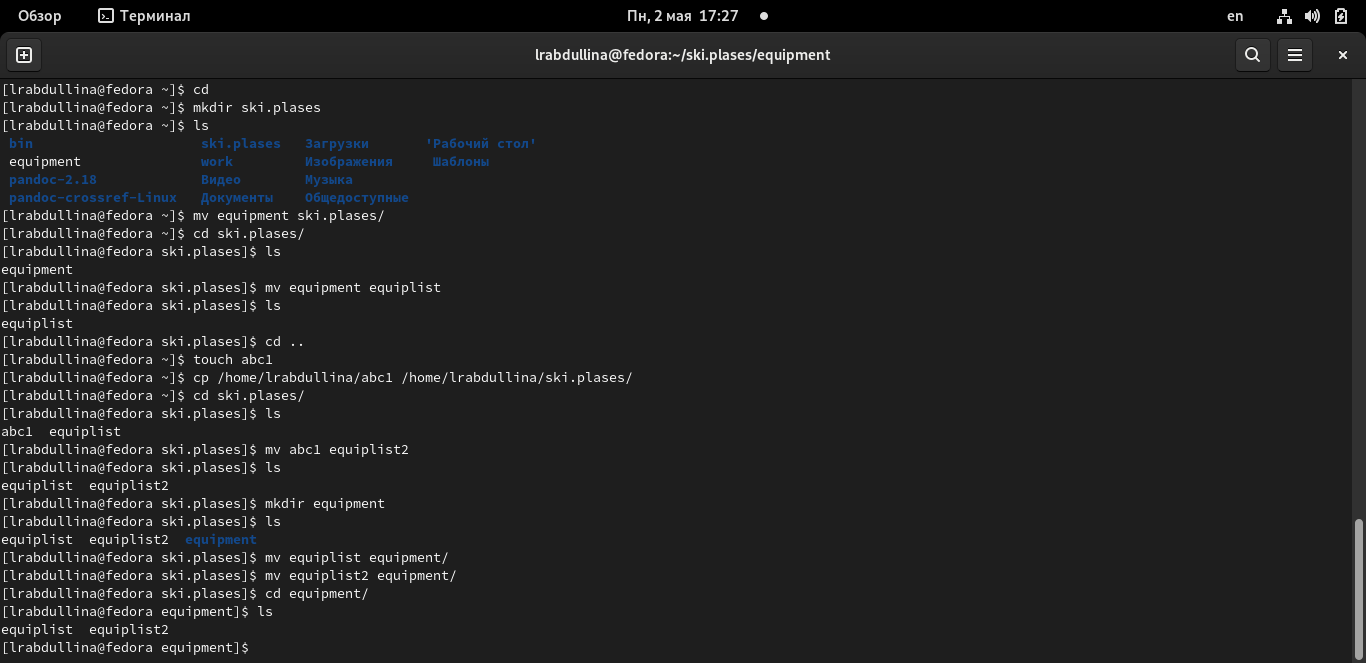
{#fig:007width=100%}

1. Создадим каталог с именем equipment в каталоге ~/ski.plases через mkdir. (Скриншот 7)



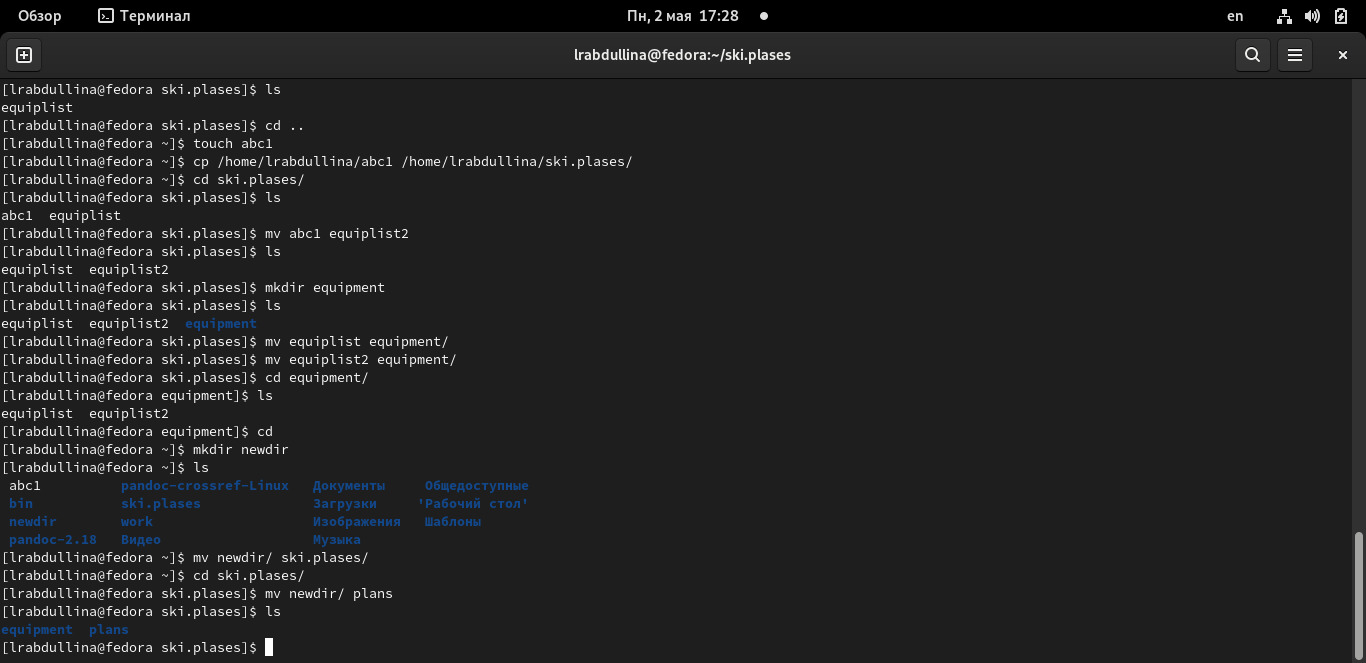
Скриншот 7: Cоздание каталога, проверка

1. Переместим файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment через команду mv. (Скриншот 8)



Скриншот 8: Перемещение, проверка

1. Создайдим (через команду mkdir) и переместим каталог ~/newdir в каталог ~/ski.plases и назовите его plans с помощью команды mv. (Скриншот 9)

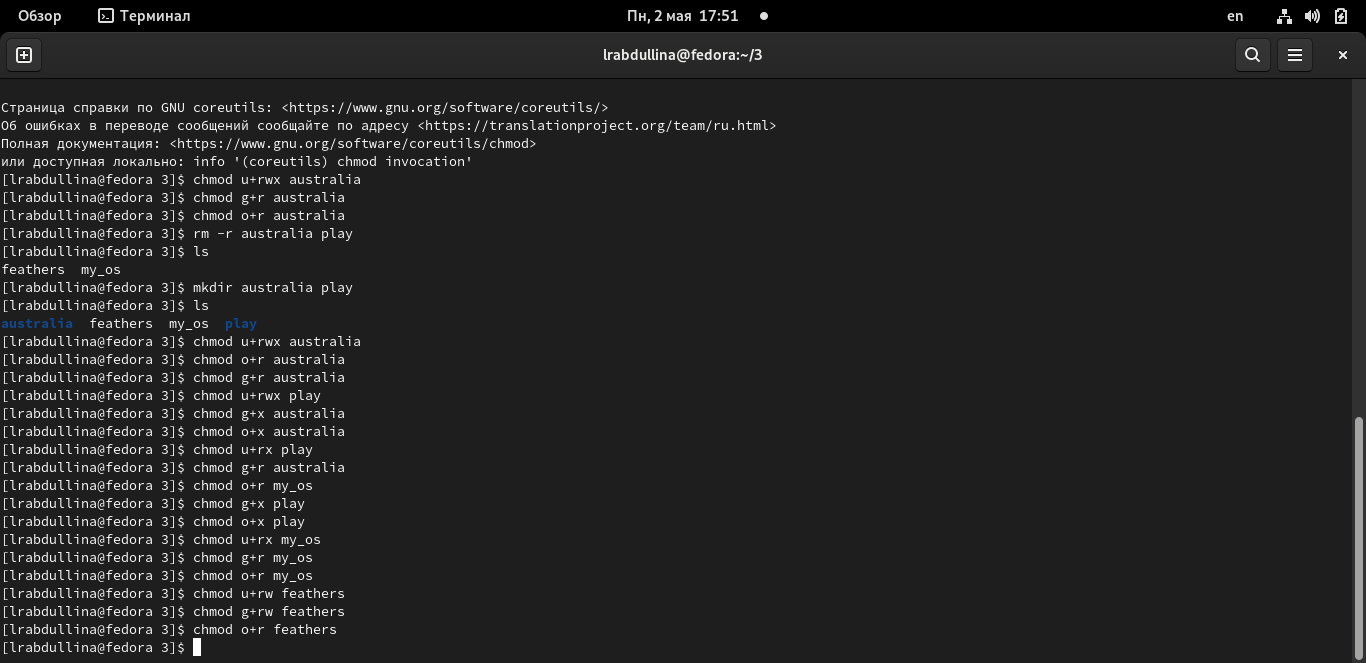


Скриншот 9: Создание нового каталога, перемещение, переименовывание, проверка

##3

Определим опции команды chmod, необходимые для того, чтобы присвоить перечисленным ниже файлам и каталогам выделенные права доступа, считая, что в начале таких прав нет. Но для начала создадим их (Скриншот 10):

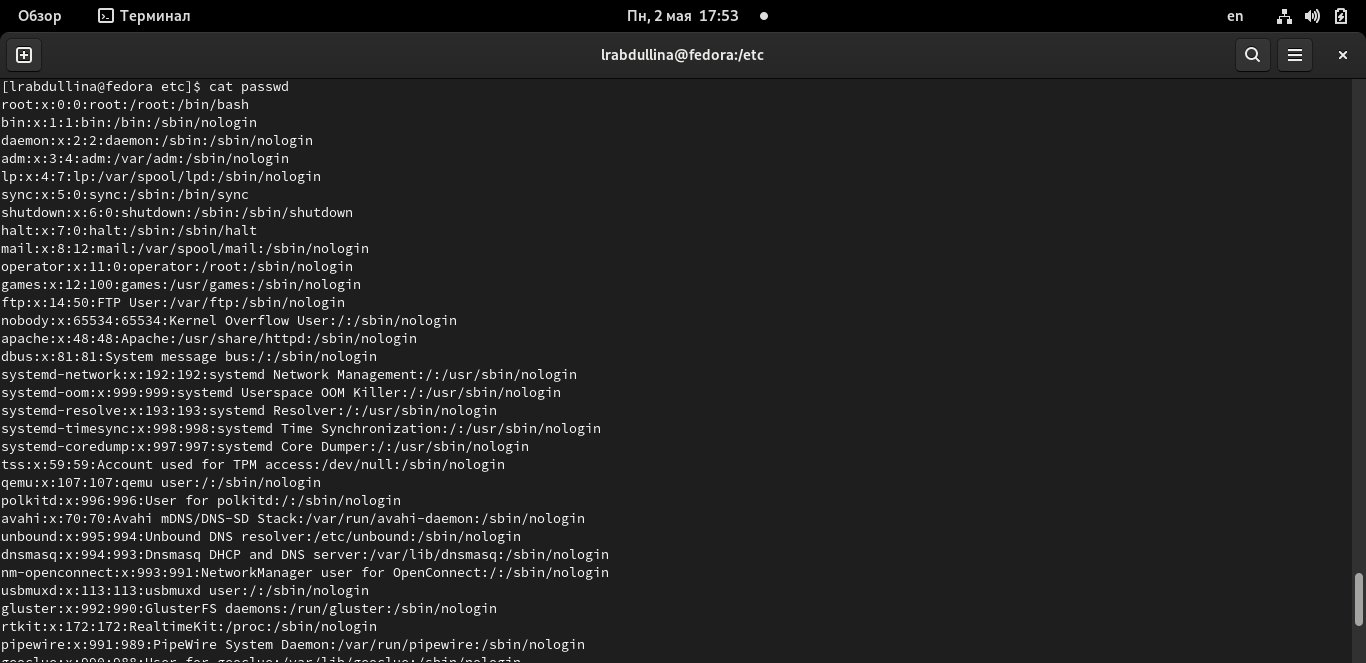
1. drwxr–r– (u+rwx, g+r, o+r) australia - каталог
2. drwx–x–x (u+rwx, g+x, o+x) play - каталог
3. -r-xr–r– (u+rx, g+r, o+r) my\_os - файл
4. -rw-rw-r– (u+rw, g+rw, o+r) feathers - файл



Скриншот 10: Наделение каталогов и файлов необходимыми правами

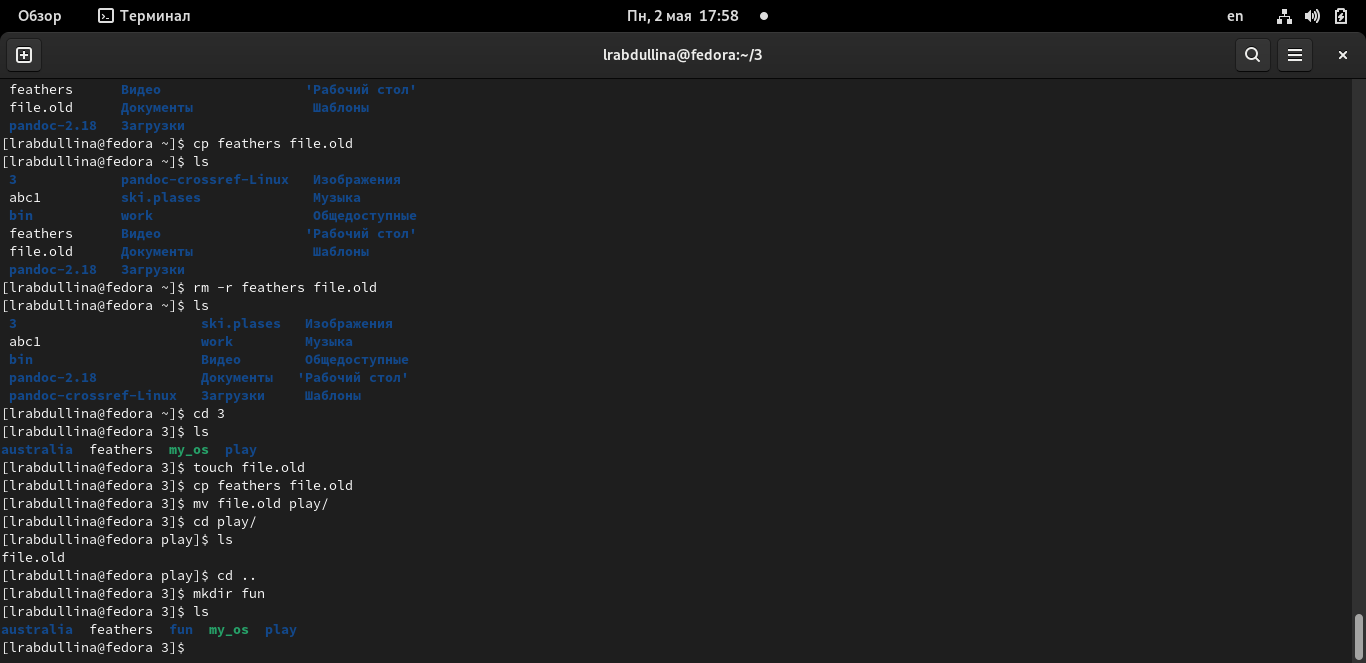
##4 Проделаем приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды.

1. Просмотрим содержимое файла /etc/passwrd с помощью ккоманды cat. (Скриншот 11).



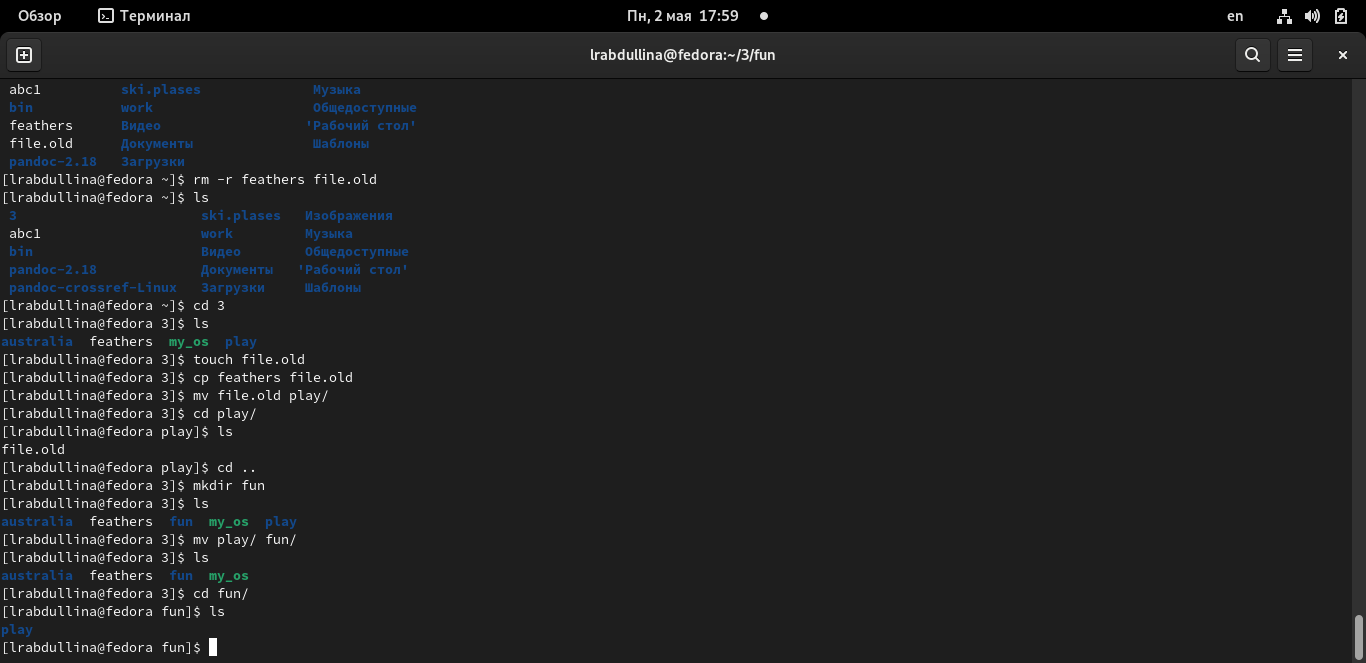
Скриншот 11: Просмотр файла

1. Скопируем файл ~/feathers в файл ~/file.old через команду cp (Скриншот 12).



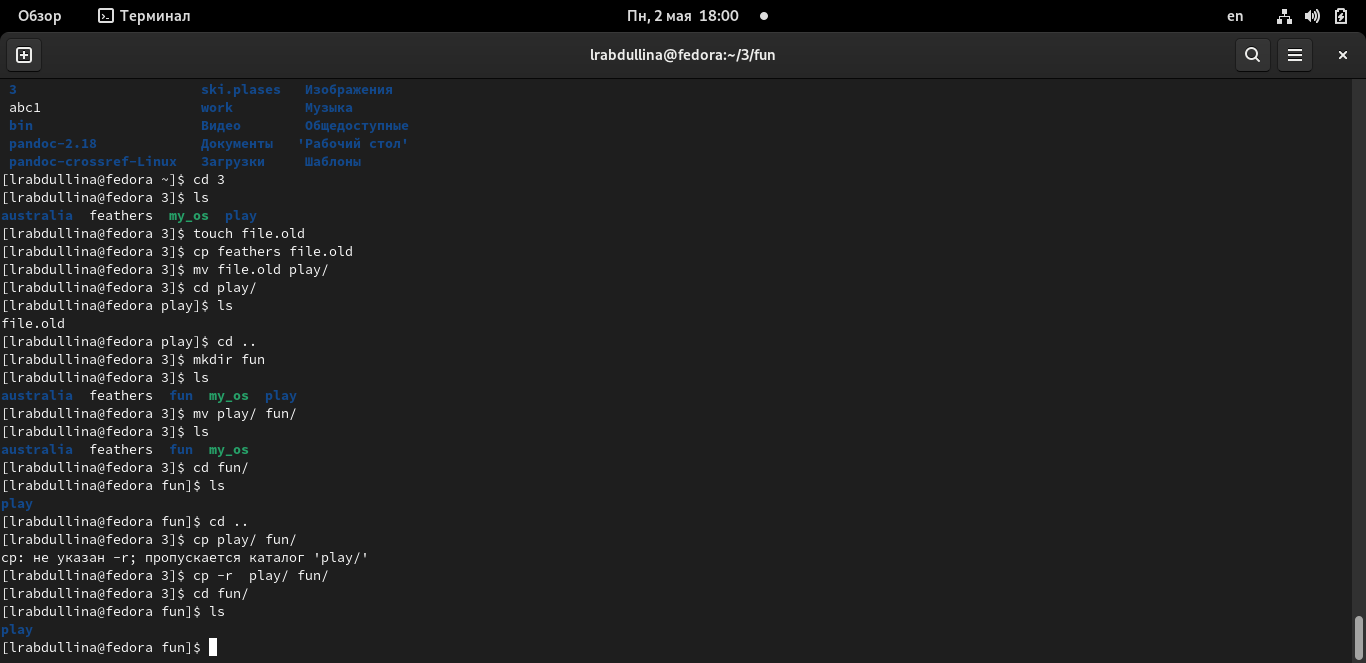
Скриншот 12: Копирование файла, проверка

1. Переместим файл ~/file.old в каталог ~/play черех команду mv(Скриншот 13).



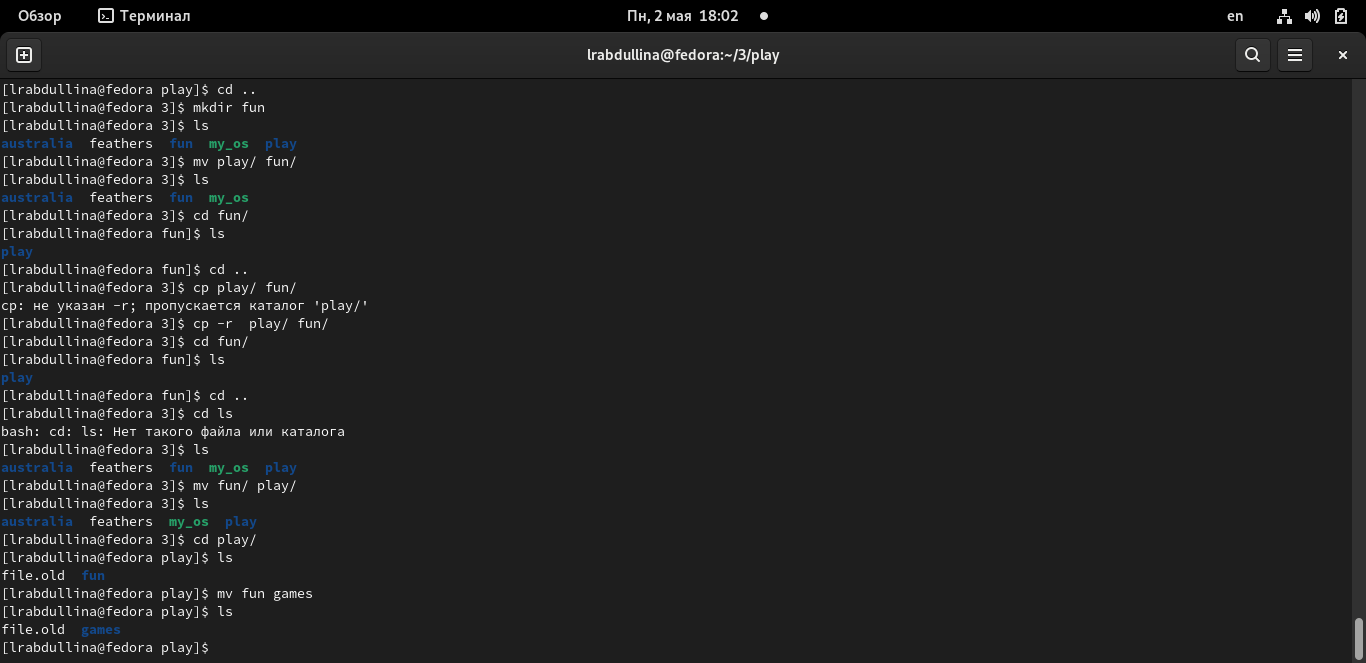
Скриншот 13: Перемещение файла, проверка

1. Скопируем каталог ~/play в каталог ~/fun через команду сp (Скриншот 14).



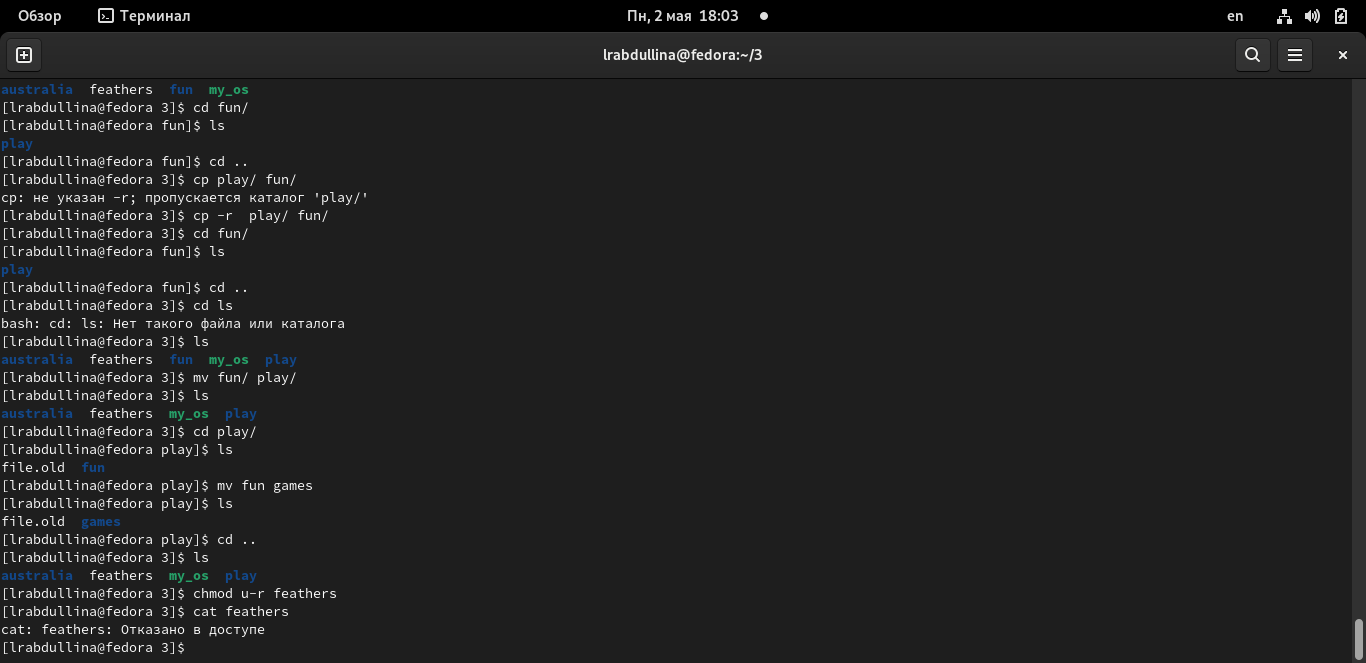
Скриншот 14: Создание каталога fun, копирование старого каталога в новый, проверка

1. Переместим каталог ~/fun в каталог ~/play и назовите его games через команду mv (поэтапно) (Скриншот 15).



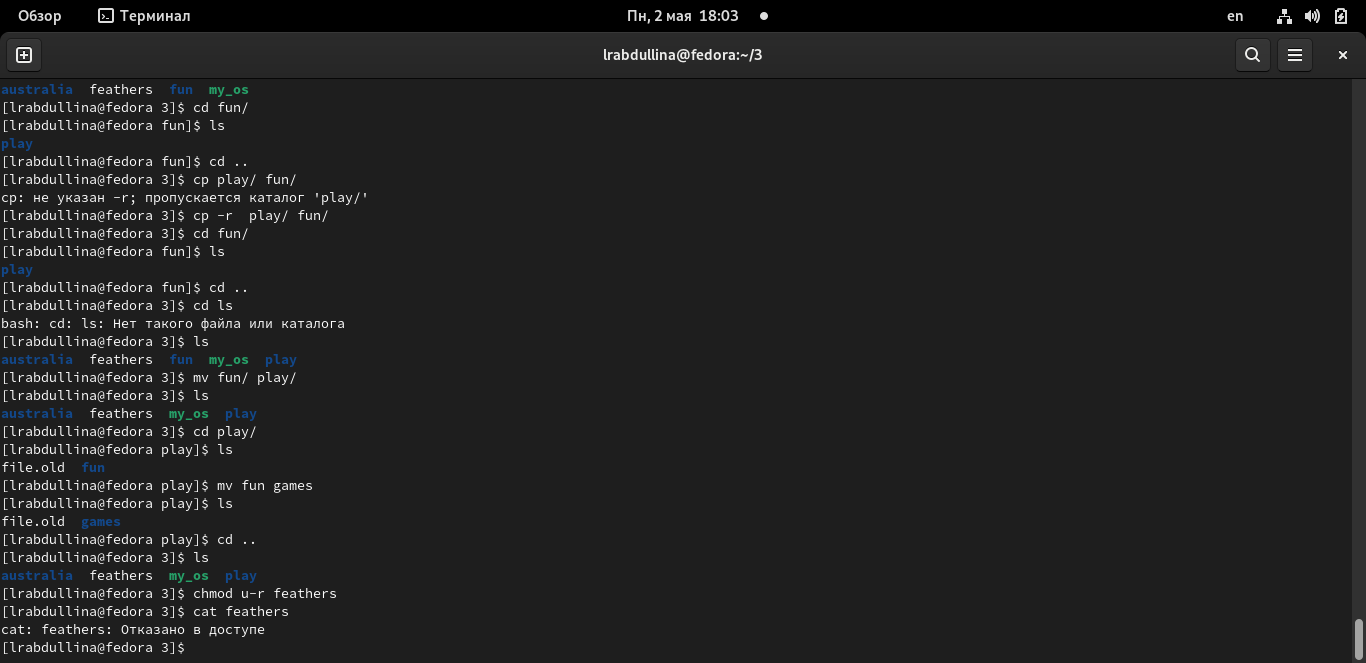
Скриншот 15: Перемещение файла, переименовывание, проверка

1. Лишим владельца файла ~/feathers права на чтение через команду chmod u-r feathers. (Скриншот 16).



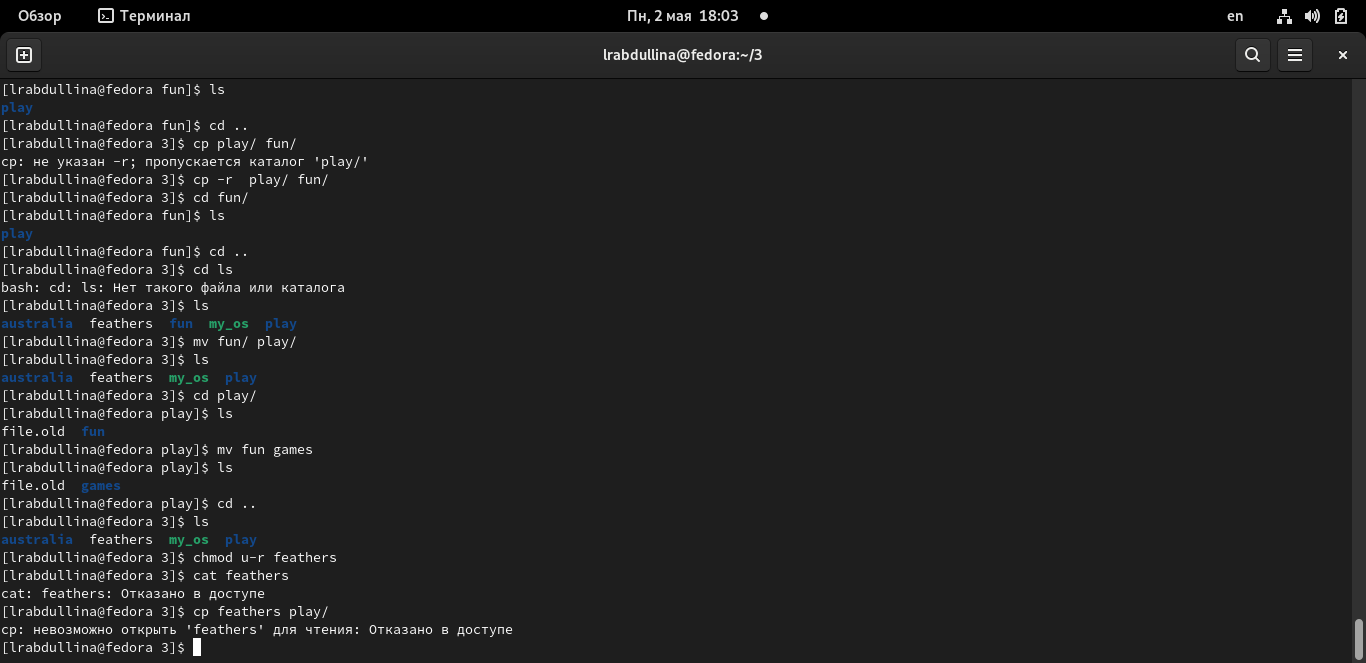
Скриншот 16: Лишение владельца прав на чтение

1. Что произойдёт,если мы попытаемся просмотреть файл ~/feathers командой cat? *Будет отказано в доступе* (Скриншот 17)



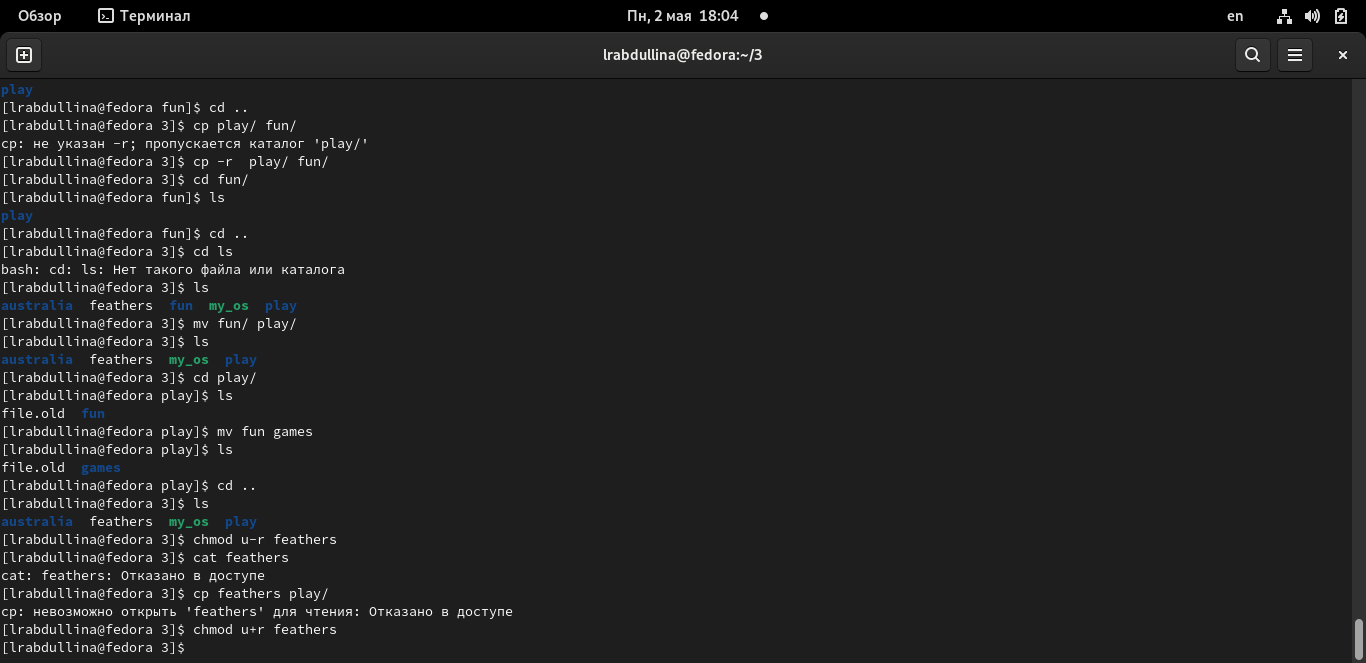
Скриншот 17: Отказ в доступе на прочтение feathers

1. Что произойдёт, если мы попытаемся скопировать файл ~/feathers? через команду cp *Ничего не получится:отказано в доступе* (Скриншот 18)



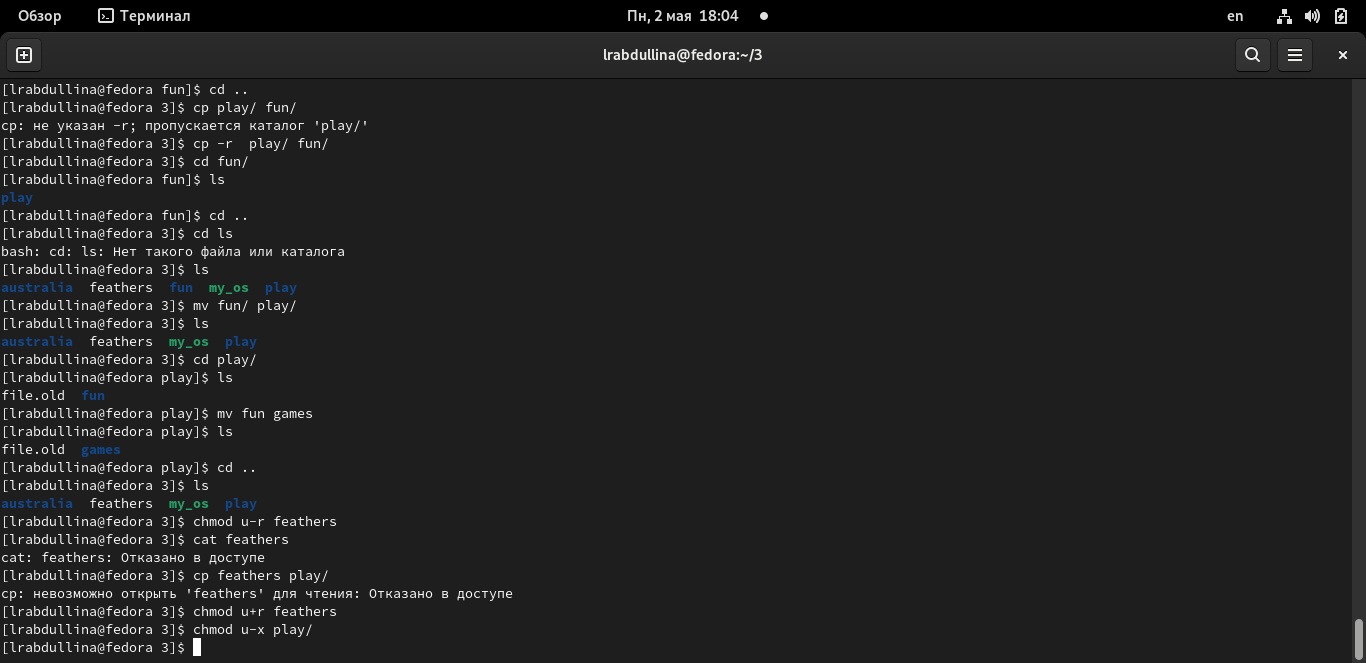
Скриншот 18: Отказ в доступе на коирование, следовательно и на прочтение feathers

1. Дадим владельцу файла ~/feathers право на чтение через команду сhmod u+r feathers. (Скриншот 19)



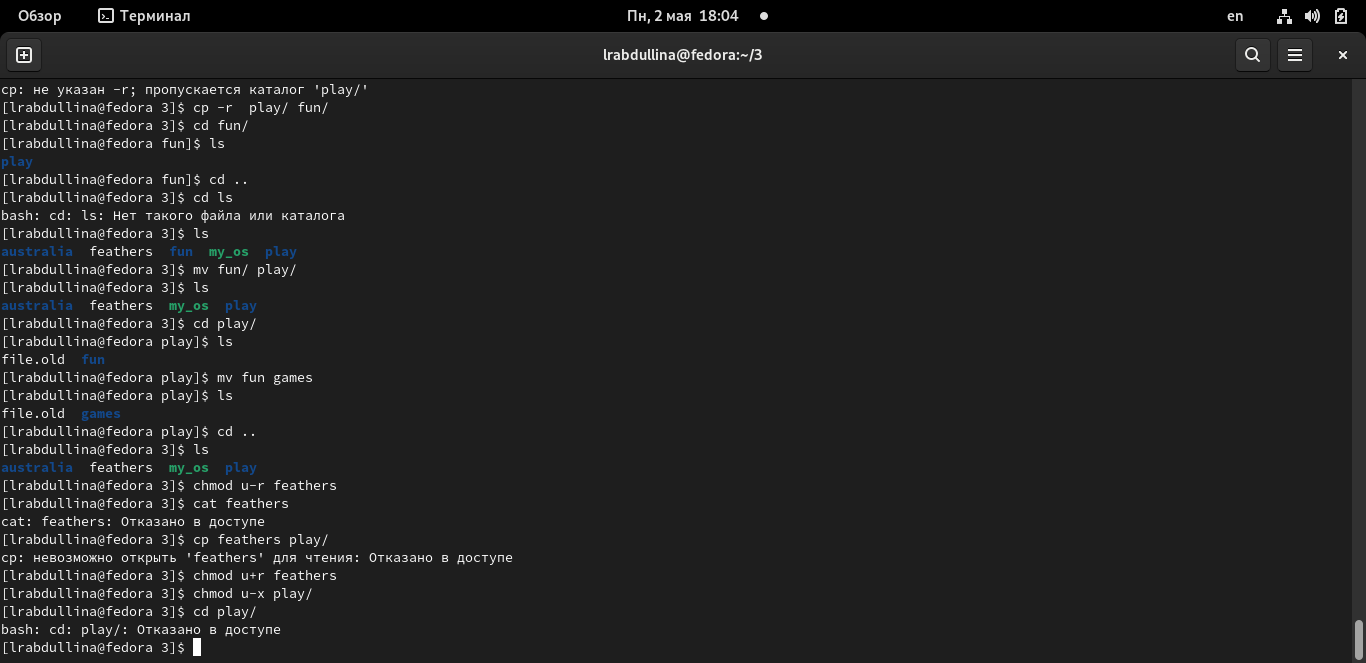
Скриншот 19: Даем права на прочтение

1. Лишим владельца каталога ~/play права на выполнение через команду chmod u-x play/.(Скриншот 20)



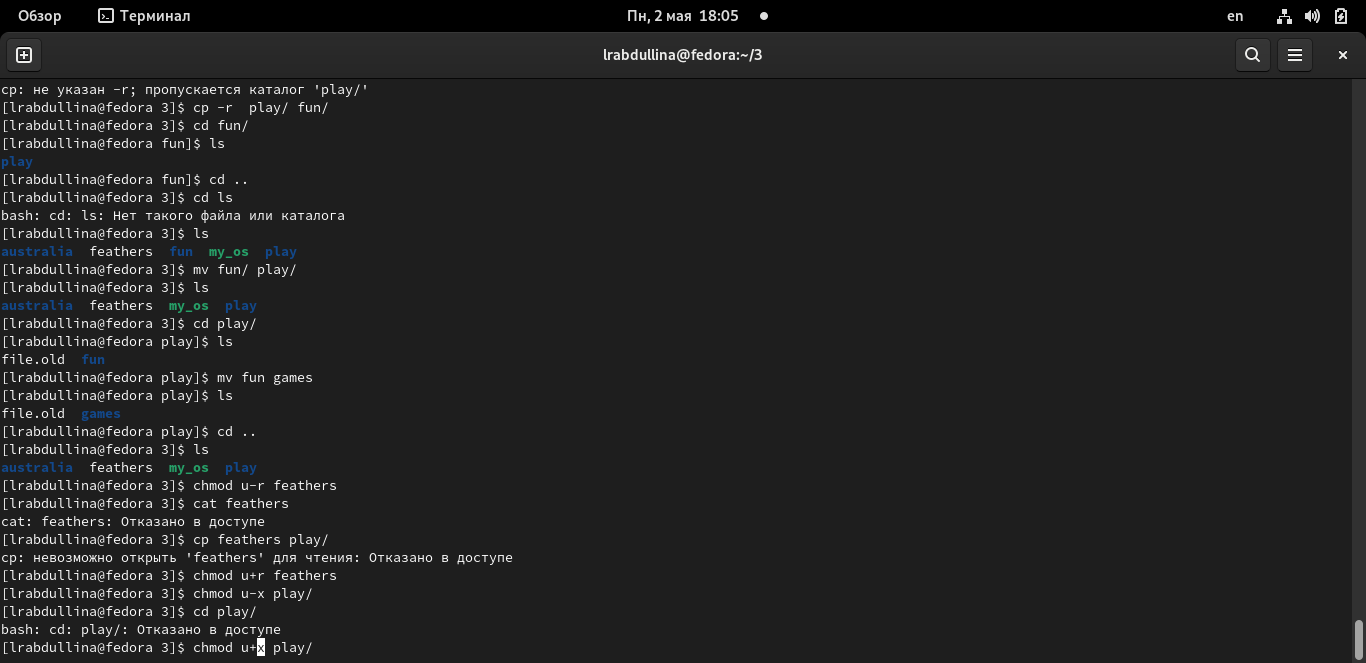
Скриншот 20: Лишение прав на выполнение

1. Перейдем в каталог ~/play. Что произошло? *Нам отказали в доступе* (Скриншот 21)



Скриншот 21: Отказ в доступе

1. Дадим владельцу каталога ~/play право на выполнение через команду chmod u+x play/. (Скриншот 22)

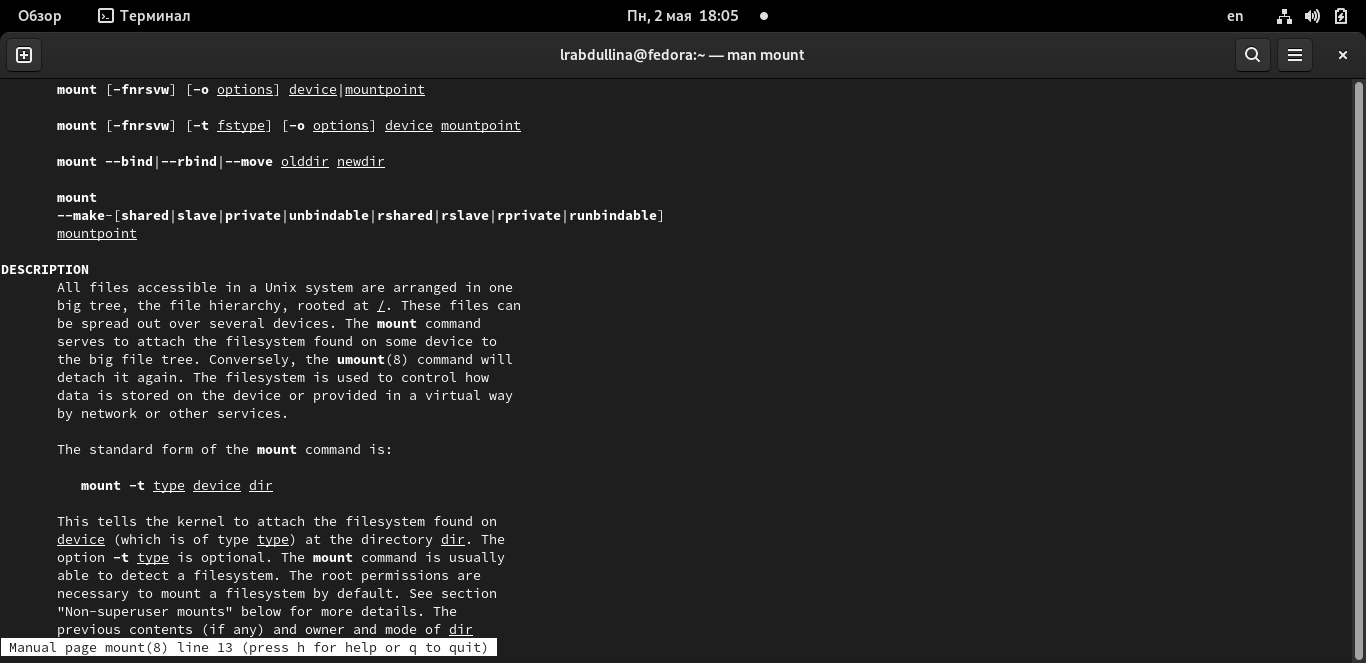


Скриншот 22: Предоставление прав

##5

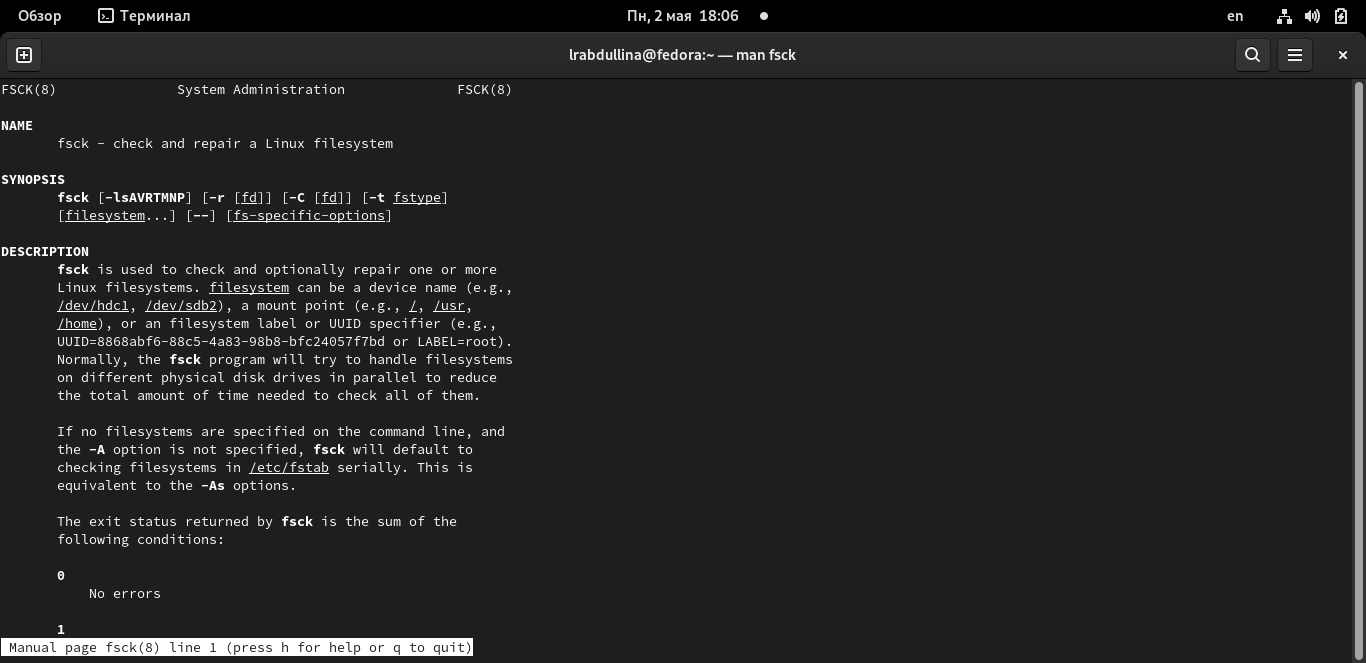
Прочитаем man по командам mount,fsck,mkfs,kill и кратко их охарактеризуем, приведя примеры. (Скриншоты 23,24,25,26)

1. *Все файлы, доступные в системе Unix расположены в одном большом “дереве”, файоловой иерархии, с корнем в /. Эти файлы могут быть распространены на несеолько разных устройств. Команда mount служит для прикрепления(монтирования) файлов, найденны на этих устройствах у этому файловому “дереву” Пример: sudo mount /dev/sda5 /mnt/win\_d*



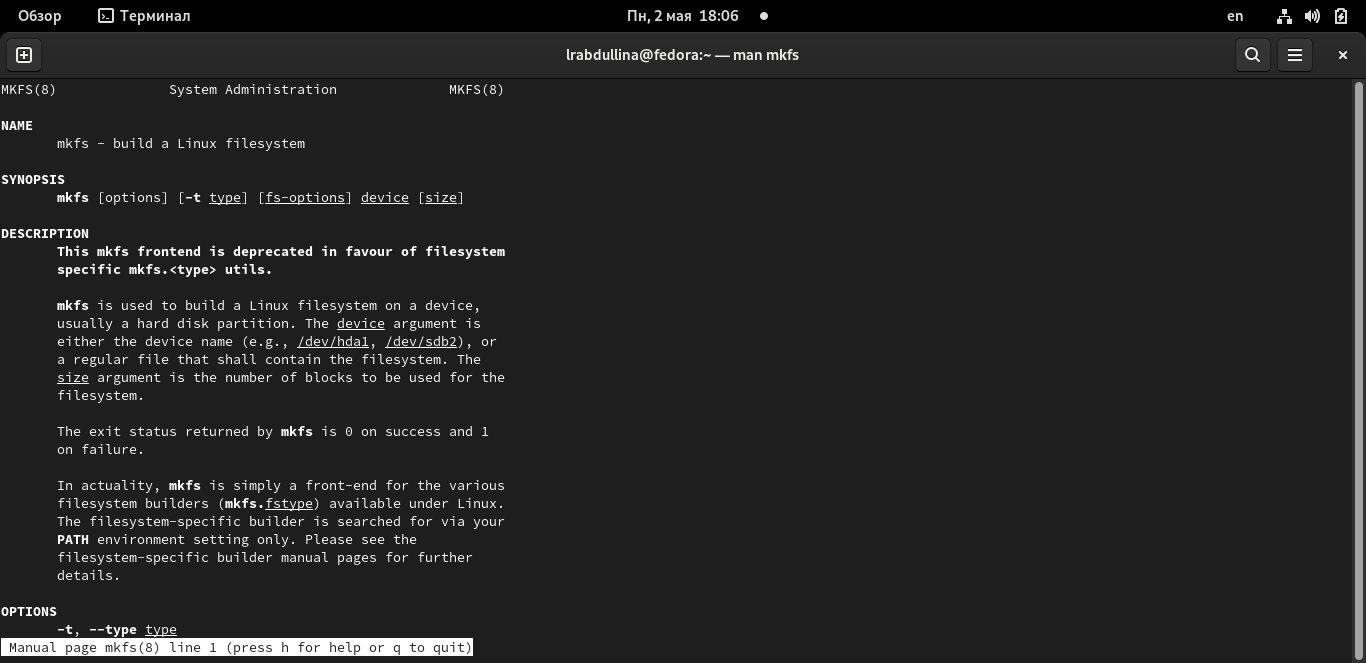
Скриншот 23: Характеристика mount

1. *Команда fsck используется для того чтобы проверять и выборочно чинить одну или несколько файловых систем Linux.Пример: udo fsck -y /dev/sdb2*



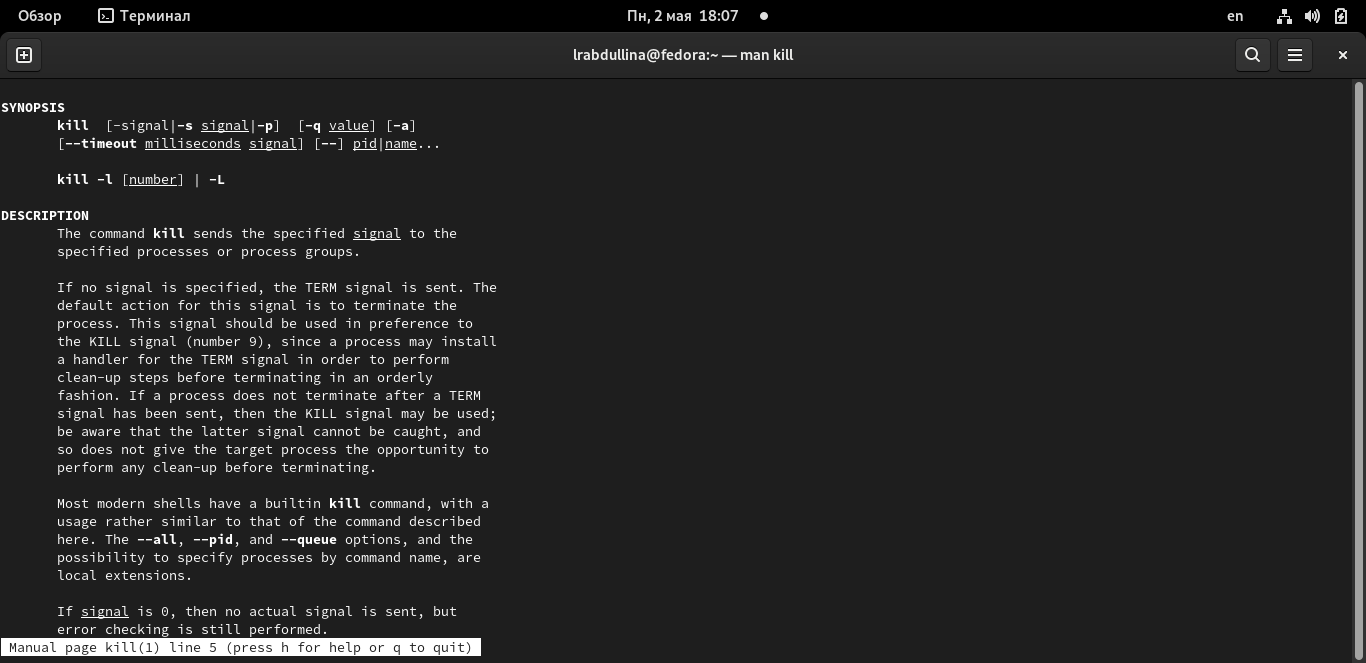
Скриншот 24: Характеристика fsck

1. *Команда mkfs используется для создания файловой системы Linux на некотором устройстве, обычно в разделе жёсткого диска. Пример: Создаёт файловую систему типа ext2 в разделе /dev/hdb1 (второй жёсткий диск): mkfs -t ext2 /dev/hdb1*



Скриншот 25: Характеристика mkfs

1. *Команда kill посылает специальный сигнал для определенных процессов (или групп процессов). В стандартном случае эта команда прерывет процесс. Пример: выполнение Makefile. При вооде команды kill - процесс остановится*



Скриншот 26: Характеристика kill

# 4 Контрольные вопросы

1. Дайте характеристику каждой файловой системе, существующей на жёстком диске компьютера, на котором вы выполняли лабораторную работу.

* **Ext2, Ext3, Ext4 или Extended Filesystem** - это стандартная файловая система для Linux. Она была разработана еще для Minix. Она самая стабильная из всех существующих, кодовая база изменяется очень редко и эта файловая система содержит больше всего функций. Версия ext2 была разработана уже именно для Linux и получила много улучшений. В 2001 году вышла ext3, которая добавила еще больше стабильности благодаря использованию журналирования. В 2006 была выпущена версия ext4, которая используется во всех дистрибутивах Linux до сегодняшнего дня. В ней было внесено много улучшений, в том числе увеличен максимальный размер раздела до одного экзабайта.
* **JFS или Journaled File System** была разработана в IBM для AIX UNIX и использовалась в качестве альтернативы для файловых систем ext. Сейчас она используется там, где необходима высокая стабильность и минимальное потребление ресурсов. При разработке файловой системы ставилась цель создать максимально эффективную файловую систему для многопроцессорных компьютеров. Также как и ext, это журналируемая файловая система, но в журнале хранятся только метаданные, что может привести к использованию старых версий файлов после сбоев.
* **ReiserFS** - была разработана намного позже, в качестве альтернативы ext3 с улучшенной производительностью и расширенными возможностями. Она была разработана под руководством Ганса Райзера и поддерживает только Linux. Из особенностей можно отметить динамический размер блока, что позволяет упаковывать несколько небольших файлов в один блок, что предотвращает фрагментацию и улучшает работу с небольшими файлами. Еще одно преимущество - в возможности изменять размеры разделов на лету. Но минус в некоторой нестабильности и риске потери данных при отключении энергии. Раньше ReiserFS применялась по умолчанию в SUSE Linux, но сейчас разработчики перешли на Btrfs.

-**XFS** - это высокопроизводительная файловая система, разработанная в Silicon Graphics для собственной операционной системы еще в 2001 году. Она изначально была рассчитана на файлы большого размера, и поддерживала диски до 2 Терабайт. Из преимуществ файловой системы можно отметить высокую скорость работы с большими файлами, отложенное выделение места, увеличение разделов на лету и незначительный размер служебной информации.

* **Btrfs или B-Tree File System** - это совершенно новая файловая система, которая сосредоточена на отказоустойчивости, легкости администрирования и восстановления данных. Файловая система объединяет в себе очень много новых интересных возможностей, таких как размещение на нескольких разделах, поддержка подтомов, изменение размера не лету, создание мгновенных снимков, а также высокая производительность. Но многими пользователями файловая система Btrfs считается нестабильной. Тем не менее, она уже используется как файловая система по умолчанию в OpenSUSE и SUSE Linux.

1. Приведите общую структуру файловой системы и дайте характеристику каждой директории первого уровня этой структуры.

* / — root каталог. Содержит в себе всю иерархию системы;
* /bin — здесь находятся двоичные исполняемые файлы. Основные общие команды, хранящиеся отдельно от других программ в системе (прим.: pwd, ls, cat, ps);
* /boot — тут расположены файлы, используемые для загрузки системы (образ initrd, ядро vmlinuz);
* /dev — в данной директории располагаются файлы устройств (драйверов). С помощью этих файлов можно взаимодействовать с устройствами. К примеру, если это жесткий диск, можно подключить его к файловой системе. В файл принтера же можно написать напрямую и отправить задание на печать;
* /etc — в этой директории находятся файлы конфигураций программ. Эти файлы позволяют настраивать системы, сервисы, скрипты системных демонов;
* /home — каталог, аналогичный каталогу Users в Windows. Содержит домашние каталоги учетных записей пользователей (кроме root). При создании нового пользователя здесь создается одноименный каталог с аналогичным именем и хранит личные файлы этого пользователя;
* /lib — содержит системные библиотеки, с которыми работают программы и модули ядра;
* /lost+found — содержит файлы, восстановленные после сбоя работы системы. Система проведет проверку после сбоя и найденные файлы можно будет посмотреть в данном каталоге;
* /media — точка монтирования внешних носителей. Например, когда вы вставляете диск в дисковод, он будет автоматически смонтирован в директорию /media/cdrom;
* /mnt — точка временного монтирования. Файловые системы подключаемых устройств обычно монтируются в этот каталог для временного использования;
* /opt — тут расположены дополнительные (необязательные) приложения. Такие программы обычно не подчиняются принятой иерархии и хранят свои файлы в одном подкаталоге (бинарные, библиотеки, конфигурации);
* /proc — содержит файлы, хранящие информацию о запущенных процессах и о состоянии ядра ОС;
* /root — директория, которая содержит файлы и личные настройки суперпользователя;
* /run — содержит файлы состояния приложений. Например, PID-файлы или UNIX-сокеты;
* /sbin — аналогично /bin содержит бинарные файлы. Утилиты нужны для настройки и администрирования системы суперпользователем;
* /srv — содержит файлы сервисов, предоставляемых сервером (прим. FTP или Apache HTTP);
* /sys — содержит данные непосредственно о системе. Тут можно узнать информацию о ядре, драйверах и устройствах;
* /tmp — содержит временные файлы. Данные файлы доступны всем пользователям на чтение и запись. Стоит отметить, что данный каталог очищается при перезагрузке;
* /usr — содержит пользовательские приложения и утилиты второго уровня, используемые пользователями, а не системой. Содержимое доступно только для чтения (кроме root). Каталог имеет вторичную иерархию и похож на корневой;
* /var — содержит переменные файлы. Имеет подкаталоги, отвечающие за отдельные переменные. Например, логи будут храниться в /var/log, кэш в /var/cache, очереди заданий в /var/spool/ и так далее.

1. Какая операция должна быть выполнена, чтобы содержимое некоторой файловой системы было доступно операционной системе?

Монтирование тома.

1. Назовите основные причины нарушения целостности файловой системы. Как устранить повреждения файловой системы?

Отсутствие синхронизации между образом файловой системы в памяти и ее данными на диске в случае аварийного останова может привести к появлению следующих ошибок:

* Один блок адресуется несколькими mode (принадлежит нескольким файлам).
* Блок помечен как свободный, но в то же время занят (на него ссылается onode).
* Блок помечен как занятый, но в то же время свободен (ни один inode на него не ссылается).
* Неправильное число ссылок в inode (недостаток или избыток ссылающихся записей в каталогах).
* Несовпадение между размером файла и суммарным размером адресуемых inode блоков.
* Недопустимые адресуемые блоки (например, расположенные за пределами файловой системы).
* “Потерянные” файлы (правильные inode, на которые не ссылаются записи каталогов).
* Недопустимые или неразмещенные номера inode в записях каталогов.

1. Как создаётся файловая система?

mkfs - позволяет создать файловую систему Linux.

1. Дайте характеристику командам для просмотра текстовых файлов.

Cat - выводит содержимое файла на стандартное устройство вывода

1. Приведите основные возможности команды cp в Linux.

Cp – копирует или перемещает директорию, файлы.

1. Приведите основные возможности команды mv в Linux.

Mv - переименовать или переместить файл или директорию

1. Что такое права доступа? Как они могут быть изменены?

Права доступа к файлу или каталогу можно изменить, воспользовавшись командой chmod. Сделать это может владелец файла (или каталога) или пользователь с правами администратора.

# 5 Выводы

В ходе лабораторной работы мы ознакомилмсь с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобрели практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

# 6 Список литературы

https://esystem.rudn.ru/course/view.php?id=5790 :::