РУДН. Операционные системы

Отчёт по лабораторной работе №5

Косинов Никита Андреевич, НПМбв-02-20

Содержание

# 1 Цель работы

Файловая структура - основной инструмент навигации на компьютере. Он упрощает поиск, хранение, работу с данными. Изменяя права доступа, можно защитить файлы от фатального изменения, или данные пользователя.

Цель данной работы - приобретение теоретических и практических навыков по работе с файловой ситемой из командной строки на примере ОС **Linux**.

# 2 Ход работы

Лабораторная работа выполнена в терминале **OC Linux** и хостинге хранения проектов **Github**. Действия по лабораторной работе представлены в следующем порядке: 1. примеры создания и чтения файлов из командной строки; 2. работа с файловой системой; 3. определение и изменение прав доступа; 4. самостоятельная работа.

По завершении отчёта, вся рабочая папка отправляется на репозиторий на *github*.

# 3 Примеры создания и чтения файлов из командной строки

Перед началом работы, по обычаю, получим обновления из **git** командой *git pull*, и создадим рабочую папку **lab02-05** командой *mkdir*.

Выполняем примеры из лабораторной работы.

1. Создаём новый документ **file** командой *touch*. Проверяем, что документ появился командой *ls*.

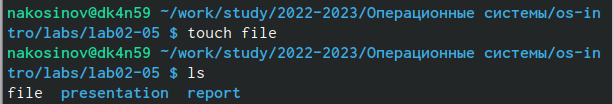


Рис. 1: Создание нового файла

1. Запишем что-нибудь в файл для дальнейших экспериментов.

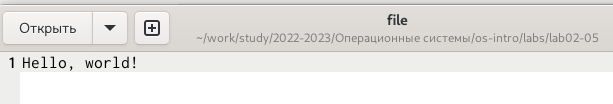


Рис. 2: Открытый документ с записью

1. Просмотирм содержимое командой *cat*.

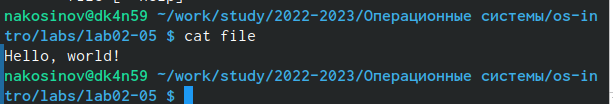


Рис. 3: Просмотр содержимого документа

1. Просмотрим содержимое документа командой *less*.

Просмотр содержимого документа

Рис. 4: Просмотр содержимого документа

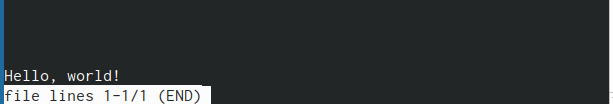


Рис. 5: Просмотр содержимого документа

1. Немного подредактировав документ **file**, выведем его первые 3 и последние 3 строки.

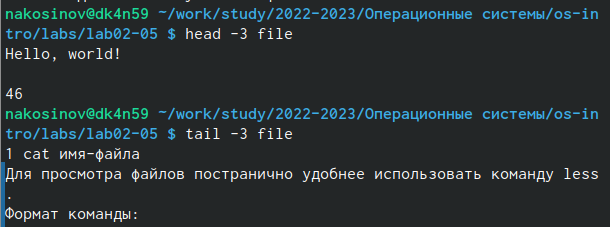


Рис. 6: Просмотр начала и конца документа

# 4 Работа с файловой системой

Данная часть работы посвящена редактированию файловой структуры посредством консоли.

1. Копируем командой *cp* файл **io.h** в текущую рабочую директорию, меняя его название на **equipment**.

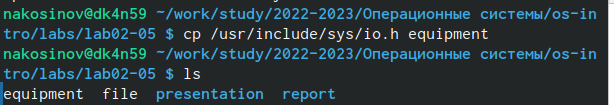


Рис. 7: Копирование файла

1. Создаём новую папку в текущем каталоге командой *mkdir*.

Создание каталога

Рис. 8: Создание каталога

1. Перемещаем ранее скопированный файл **equipment** в новый каталог с помощью команды *mv*.

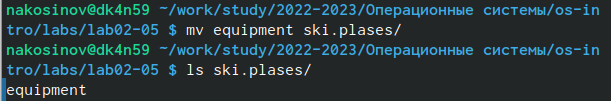


Рис. 9: Перемещение файла

1. Переименуем этот файл в **equiplist**, используя ту же команду *mv*.

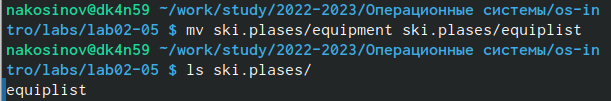


Рис. 10: Переименование файла

1. Создадим новый файл **abc1** командой *touch*. Скопируем его командой *cp* в каталог **ski.plases**, одновременно переименовывая в **equiplist2**. Сразу проверим корректность выполнения командой *ls*.

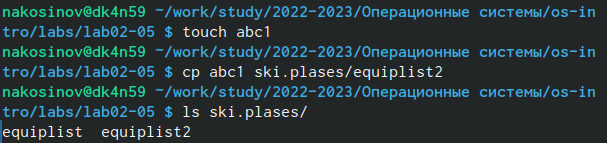


Рис. 11: Создание и перемещение файла

1. Создадим новый каталог **equipment** командой *mkdir* с указанием относительного пути.

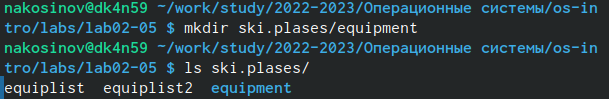


Рис. 12: Создание каталога с указанием пути

1. Переместим файлы из **ski.plases** в новую директорию одной командой *mv* с последовательным перечислением.

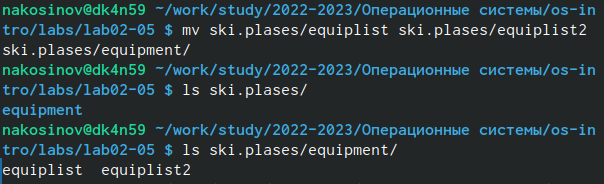


Рис. 13: Перемещение нескольких файлов с указанием пути

1. Создадим новый каталог **newdir**. Переместим его внутрь другого каталога.

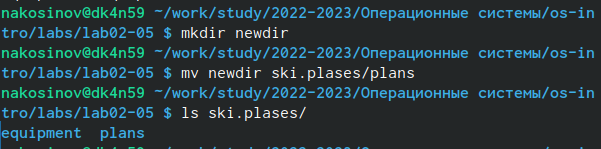


Рис. 14: Перемещение каталога

# 5 Определение и изменение прав доступа

У каждого объекта: файла или каталога, есть определённая настройка прав доступа. Различают три вида пользователей: **u** - владелец файла, **g** - группа владельца, **o** - все остальные. Для каждого вида отдельно в строчку указывается, что он может, а что не может делать с объектом: читать, записывать и выполнять.

1. Создаём новые документы **file** командой *touch* для дальнейших эксперементов.



Рис. 15: Создание новых файлов

1. Посмотрим текущие права доступа файла **australia**. Видим строчку *rw-r–r–*, что означет, что у **u** есть право на чтение и запись, у **g** и **o** - только на чтение. Добавим владельцу файла право на выполнение объекта - **x**.

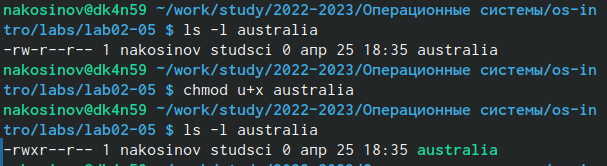


Рис. 16: Просмотр и изменение прав доступа

1. Аналогично поступим с документами **play**, **my\_os** и **feathers**. Например, для документа **play** добавим всем право на исполниение, но отменим *группе* и *остальным* право на чтение.

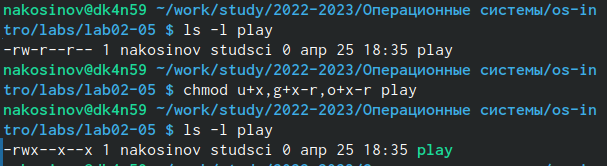


Рис. 17: Просмотр и изменение прав доступа

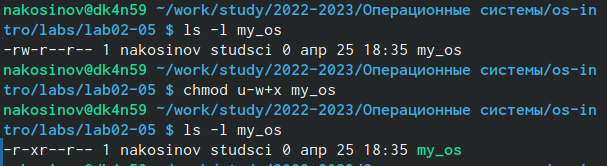


Рис. 18: Просмотр и изменение прав доступа

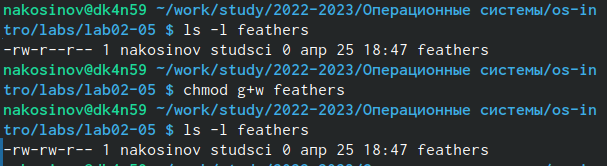


Рис. 19: Просмотр и изменение прав доступа

# 6 Самостоятельная работа

Выполним несколько упражнений самостоятельно.

1. Командой *cat* росмотрим содержимое файла **passwd** из каталога **/etc**.

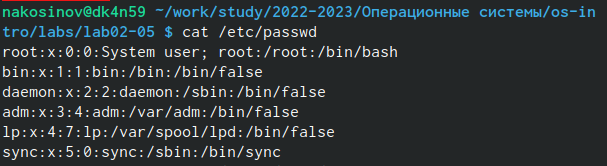


Рис. 20: Просмотр содержимого файла

1. Командой *cp* скопируем файл, оставив в той же директории.

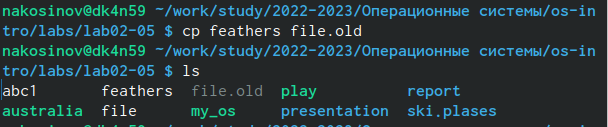


Рис. 21: Копирование файла

1. Командой *mv* переместим скопированный файл в другой каталог.

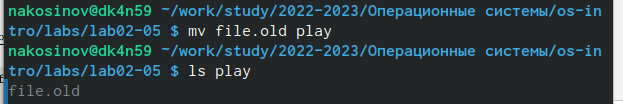


Рис. 22: Перемещение файла без изменения имени

1. Командой *cp* с использованием ключа **-r** скопируем каталог **play** со всем его содержимым.

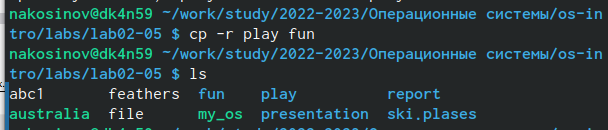


Рис. 23: Копирование каталога

1. Командой *mv* перемещаем скопированный каталог с изменением имени.

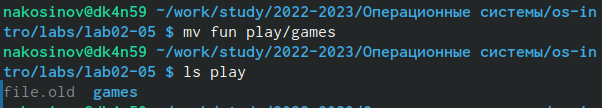


Рис. 24: Перемещение каталога с изменением имени

1. Командой *chmod* с опцией *u-r* лишаем владельца файла прав на чтение. Проверяем корректность работы командой *ls* с ключом *-l*.

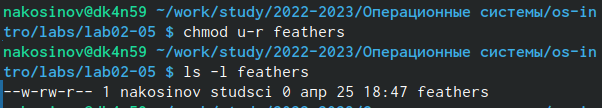


Рис. 25: Изменение прав доступа

1. Если попытаться просмотреть содержимое командой *cat* файл с запретом на чтение, то просмотреть не удастся. То же и с попыткой копирования командой *cp*.

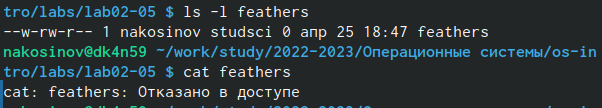


Рис. 26: Отказ в доступе к файлу с запретом на стение

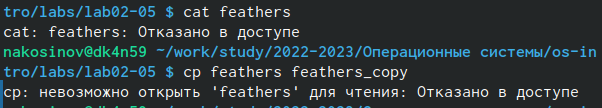


Рис. 27: Отказ в доступе к файлу с запретом на стение

1. Командой *chmod* с опцией *u+r* вернём владельцу файла права на чтение.

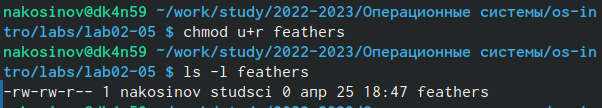


Рис. 28: Изменение прав доступа

1. Лишим владельца прав доступа на выполнение каталога **play**. В этом случае при попытке перейти в каталог, попытка увенчается успехом, ведь права на чтение сохранены.

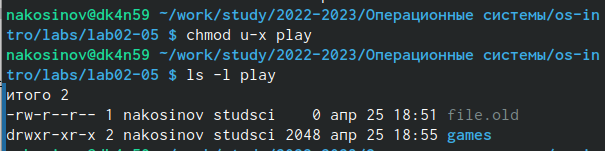


Рис. 29: Изменение прав доступа

Изменение прав доступа

Рис. 30: Изменение прав доступа

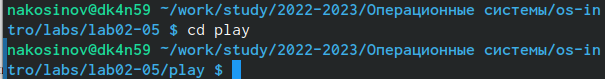


Рис. 31: Переход в каталог с закрытыми на выполнение правами

1. Просмотрим некоторые команды с помощью *man*. Из описанного там можно сделать выводы, что:

* команда **mount** позволяет изучить используемые файловые системы и примонтированные устройства;
* команда **fsck** позволяет мониторить корректность работы файловой ситемы;
* команда **mkfs** используется для создания новой файловой системы на хард-диске или флеш-накопителе;
* команда **kill** завершет исполняемые процессы, не зависимо от стадии их работы, что позволяет, например, разгрузить эвм.

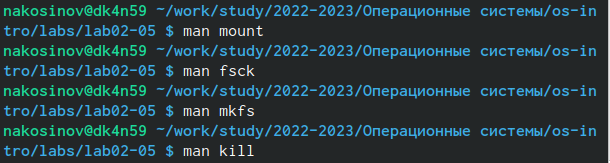


Рис. 32: Чтение мануала команд

# 7 Выводы

Терминал предоставляет пользователю возможность выполнения различных операций по управлению файловой системой: создание, перемещение объектов, изменение их прав доступа, просмотр общей информации о ФС.