МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №3**

**Дисциплина: Операционные системы**

**Тема: «РАБОТА С ФАЙЛАМИ И КАТАЛОГАМИ. FHS»**

Работу выполнил:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ахвердян Г.А

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Направленность (профиль): Математическое и программное

обеспечение компьютерных технологий

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. А. Нигодин

Краснодар

2025

С помощью команды «ls -l /home/sa/hello.txt» мы можем вывести данные о файле, такие как: права на файл, количество жестких ссылок на файл, пользователь являющимся владельцом файла, группа являющаяся владельцом, размер файла в байтах, дата последнего изменения файла, полный путь к файлу

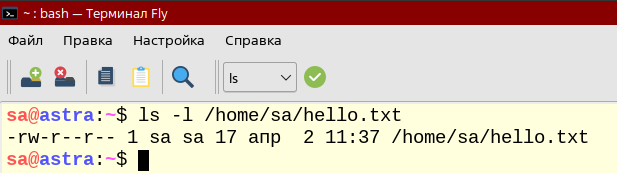


Рисунок 1 – Вывод данных о файле

Так же, можно узнать номер индексного дескриптора. Для этого нужно использовать команду «ls -i /home/sa/hello.txt»

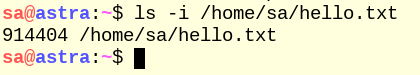


Рисунок 2 – Номер индексного дескриптора

Используя команду «stat /home/sa/hello.txt», можно получить ряд дополнительных атрибутов файла, хранящихся в его айноде. Одним из преимуществ команды stat в сравнении с ls является то, что она показывает права доступа к файлу в числовом формате, например, в приведенном ниже примере на файл hello.txt установлены права 0644

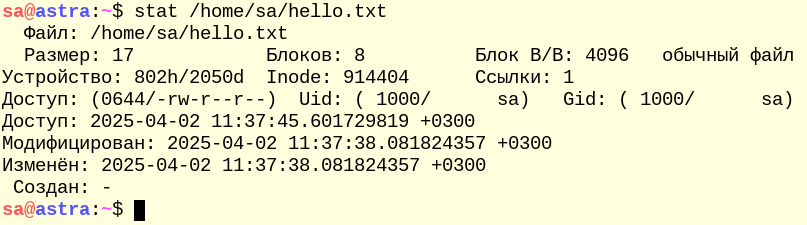


Рисунок 3 – Дополнительные атрибуты файла

В случае, если нам необходимо узнать о количестве занятых\свободных айнод, мы можем воспользоваться командой «df –i»

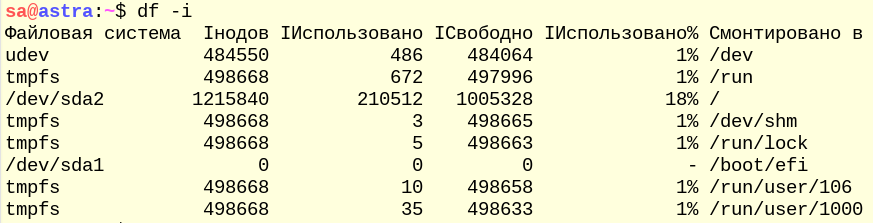


Рисунок 4 – Количество используемых айнод

К обычным файлам можно отнести все файлы с данными, например, тексты, картинки, архивы, скрипты, исполняемые файлы, библиотеки и т. д. Кратким обозначением обычных файлов является символ дефиса «-». В выводе команды мы можем увидеть несколько типов файлов, что можно определить по цвету строки и первому символу. Символьные ссылки выделяются бирюзовым цветом, каталоги - синим, исполнимые файлы - зеленым, а обычные файлы отображаются цветом по умолчанию.

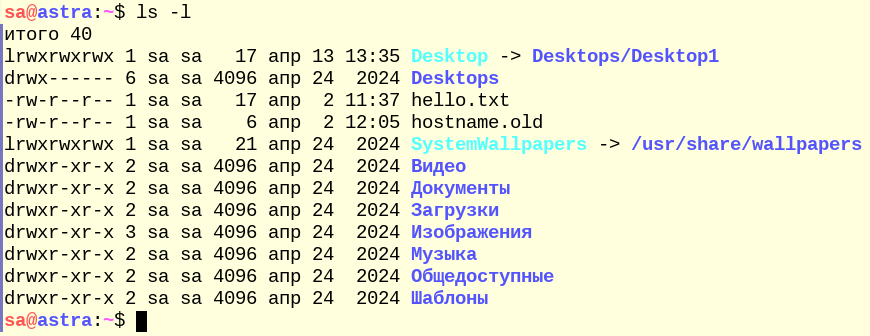


Рисунок 5 – Файлы

При создании любого файла или каталога для него создается жесткая ссылка, и счетчик ссылок становится равен единице (0 + 1 = 1). Выполним команду «mkdir test» для создания папки test, перейдем в эту папку командой «cd test» и выполним команды: «touch test1» для создания файла test1 «ls –il» для просмотра количества ссылок на объекты в текущем каталоге и номеров их индексных дескрипторов.

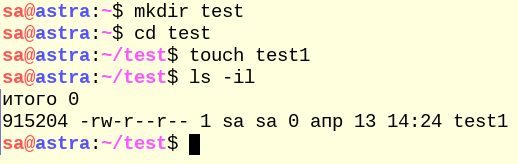


Рисунок 6 – Создание папки test с файлом test1

Теперь создадим для этого файла жесткую ссылку командой «ln test1 hardlink\_test1», и просмотрим ссылки еще раз «ls –il»

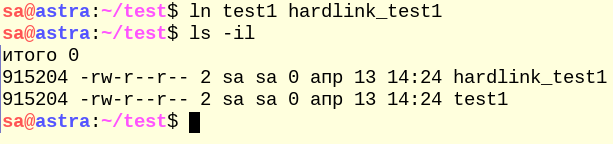


Рисунок 7 – Создание жесткой ссылки

Счетчик увеличился на единицу. Создадим подкаталог командой «mkdir subtest» и создадим в нем еще одну жёсткую ссылку на этот файл командой: «ln hardlink\_test1 subtest/hardlink2\_test1». При обращении к файлу по жесткой ссылке мы можем использовать любое имя, так как все ссылаются на одну и ту же айноду и нельзя сказать, какая из жестких ссылок была создана раньше, а какая позже. Убедиться в этом можно, выполнив команду «ls -il subtest». Теперь обновим время последнего изменения файла командой «touch hardlink\_test1» и еще раз посмотрим свойства объектов в test и subtest:

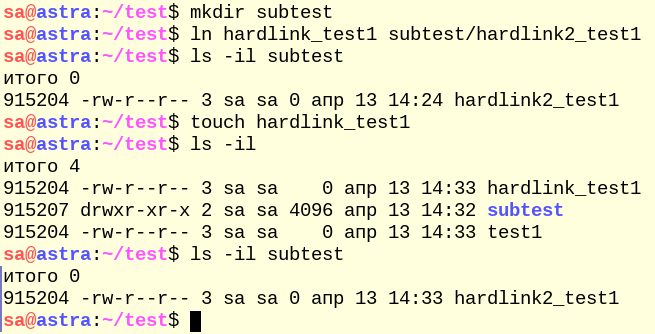


Рисунок 8 – Создание подкаталога и еще одной жесткой ссылки

Символическая ссылка ссылается на путь к файлу или папке. Если объект, на который она ссылается, будет удален (перемещен), то ссылка будет вести на несуществующий объект, то есть станет разорванной. Создадим несколько символических ссылок на файлы и папку с помощью команды ln с ключом –s. «ln -s ~/test/subtest ~/subtest» – создаем ссылку subtest в домашнем каталоге на каталог subtest в каталоге test.

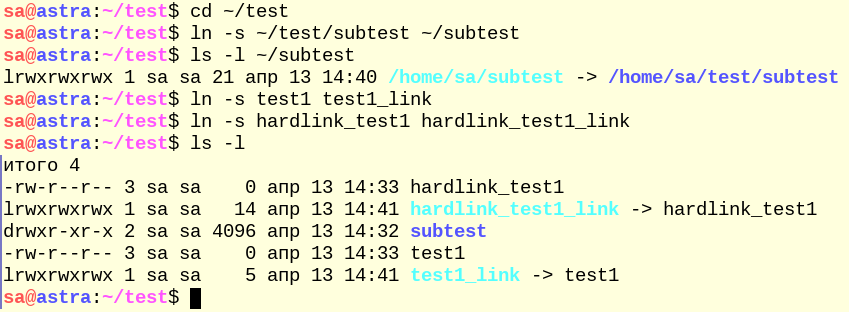


Рисунок 9 – Символические ссылки

Попробуем получить доступ к файлам и каталогам через символьные ссылки. Выполним команды «ls ~/subtest, echo "test text" > hardlink\_test1\_link» и «cat test1\_link»

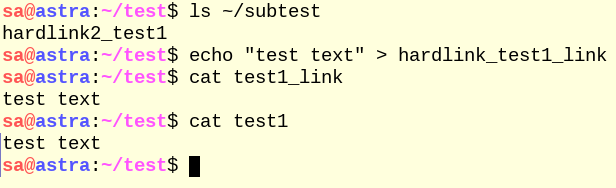


Рисунок 10 – Доступ к файлам и каталогам через символьные ссылки

Удалим жесткую ссылку hardlink\_test1 командой «rm hardlink\_test1» и проверим доступ до файла через жесткую и символическую ссылки командами «cat hardlink\_test1\_link», «cat test1\_link», «cat test1»

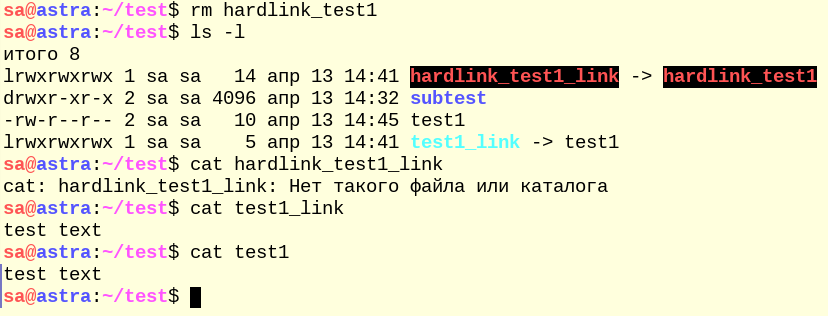


Рисунок 11 – Удаление жесткой ссылки и доступ к файлу с помощью жесткой и символической ссылками

Правила именования файлов и каталогов. Правила именования файлов и каталогов идентичны. Имя не может превышать 255 байт, а общая длина пути (включая все компоненты) не может превышать 4096 байт. Минимальный безопасный набор символов для использования в именах файлов: буквы (как латиницы, так и кириллицы в любом регистре); цифры; символ «.», символ «\_»; символ «-». Если название файла или каталога начинается с символа точки «.», то в Linux он будет считаться скрытым. Использование спецсимволов, например, пробела, символов «\*», «?», «$», «’», «”» не рекомендуется, так как их наличие в именах файлов может вызвать некорректную работу некоторых программ, хотя допустимо использовать любые символы, кроме «/». Для решения проблем со спецсимволами в именах файлов при работе в командной строке есть несколько способов: Экранирование спецсимволов. Для экранирования специальных символов перед ними нужно поставить символ обратной косой черты «\» или взять все имя файла в одинарные кавычки, например, «cat test\\ file.txt» или «cat 'test file.txt'»:

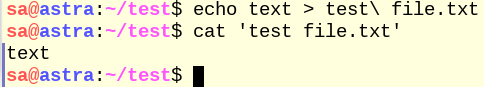


Рисунок 12 – Экранирование спецсимволов

Работа с файлами с символом тире в начале имени. Если в начале имени файла стоит символ тире -, то командный интерпретатор будет воспринимать его как параметр команды. Тут не поможет стандартное экранирование символа. Вместо него можно использовать или относительный путь к файлу, или специальный параметр --, который сообщает команде, что это не ключ, а параметр.

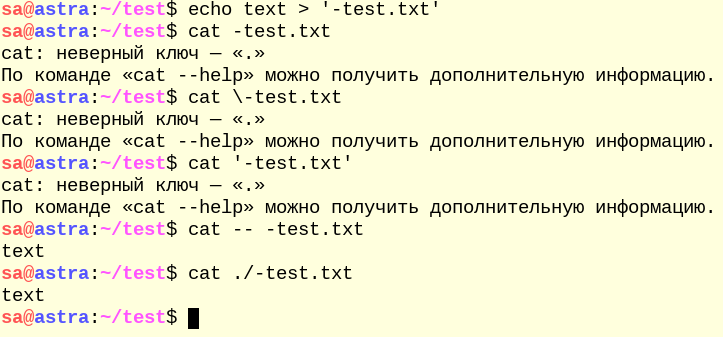


Рисунок 13 – Работа с файлами с символов тире в начале имени

Путь может быть абсолютным и относительным. Абсолютный или полный путь указывает на одно и то же место в файловой системе вне зависимости от текущего рабочего каталога или других обстоятельств, поэтому полный путь всегда начинается с корневого каталога. Относительный путь представляет собой путь по отношению к текущему рабочему каталогу пользователя или активных приложений. Символ точки «.» в начале пути означает текущий каталог, два символа точки «..» указывают, что системе для поиска файла нужно подняться на один уровень выше, символ «~» подставляет полный путь к домашнему каталогу пользователя

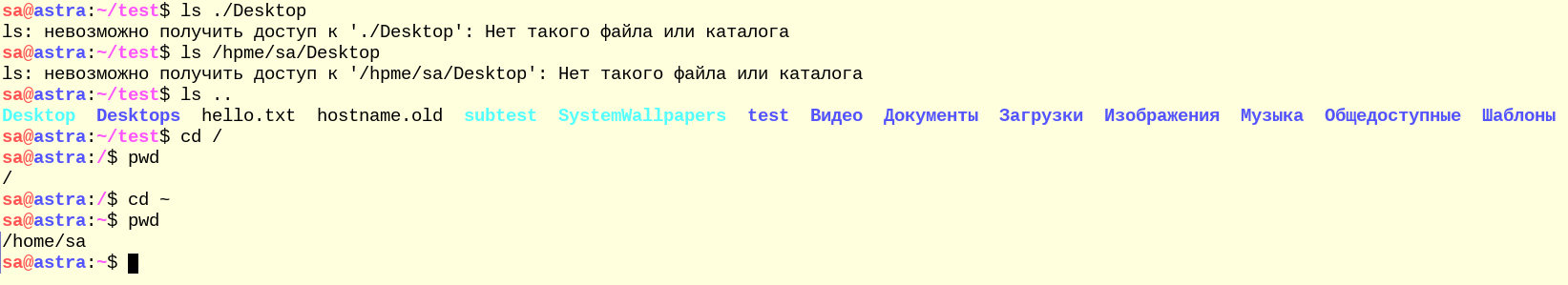


Рисунок 14 – Абсолютный и относительный путь

К основным командам можно отнести следующие: «pwd» — вывести полный путь к текущему каталогу. «cd []» — перейти в указанный каталог. «mkdir» — создать каталог. «rmdir» — удалить пустой каталог. «rm –rf» — удалить каталог со всем его содержимым. «du []» — вывести размер каталога.

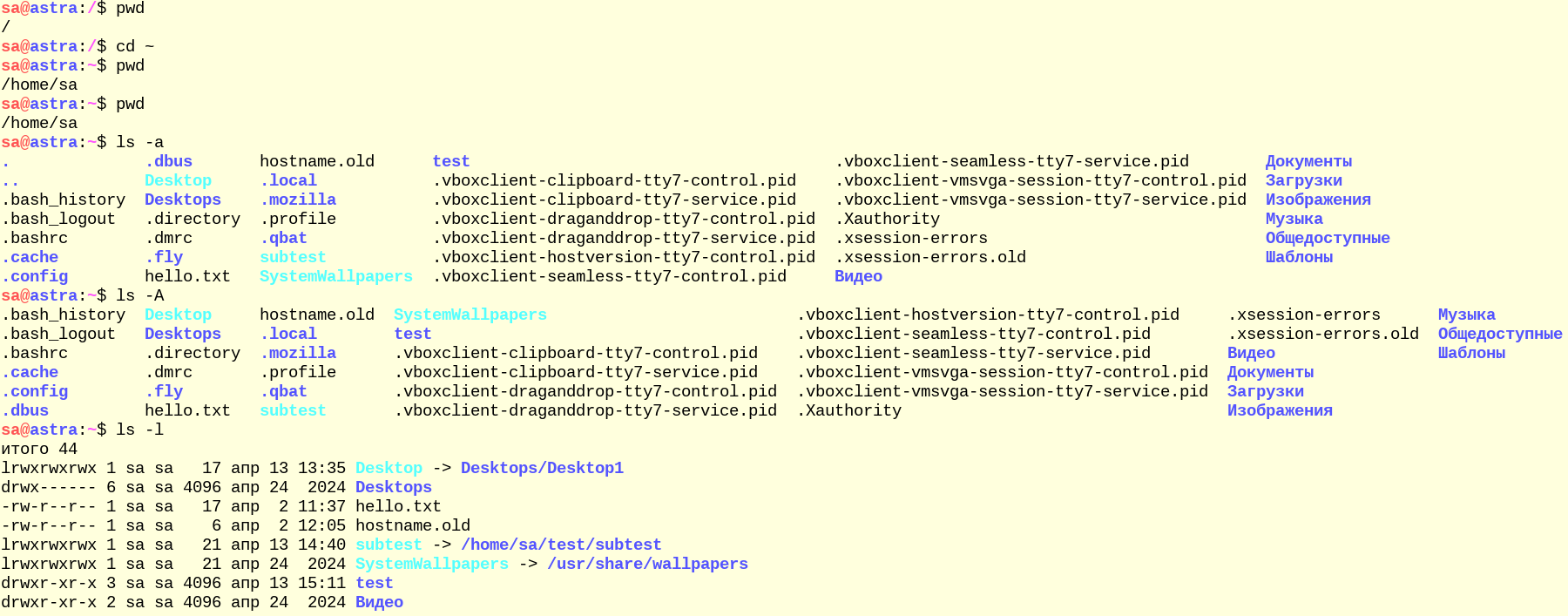


Рисунок 15 – Работа с каталогами

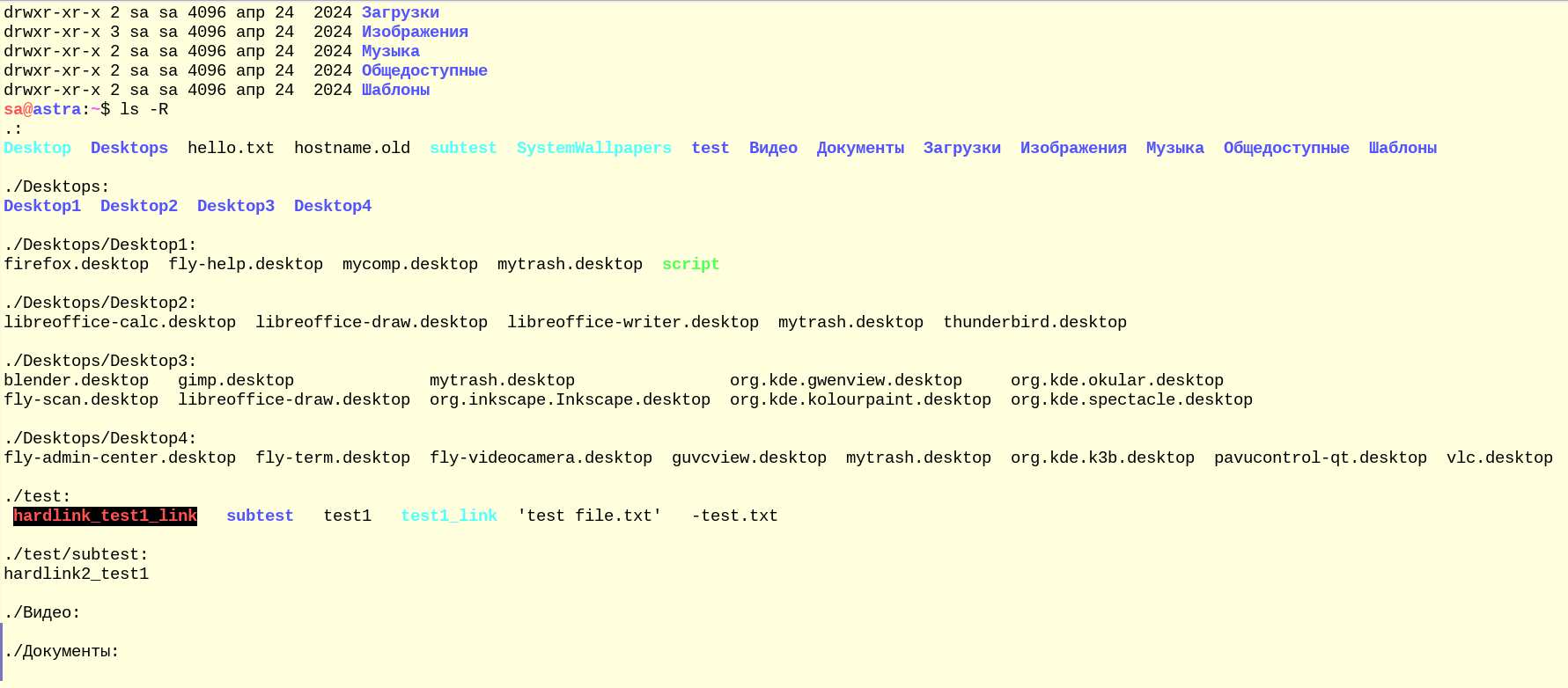


Рисунок 16 – Работа с каталогами

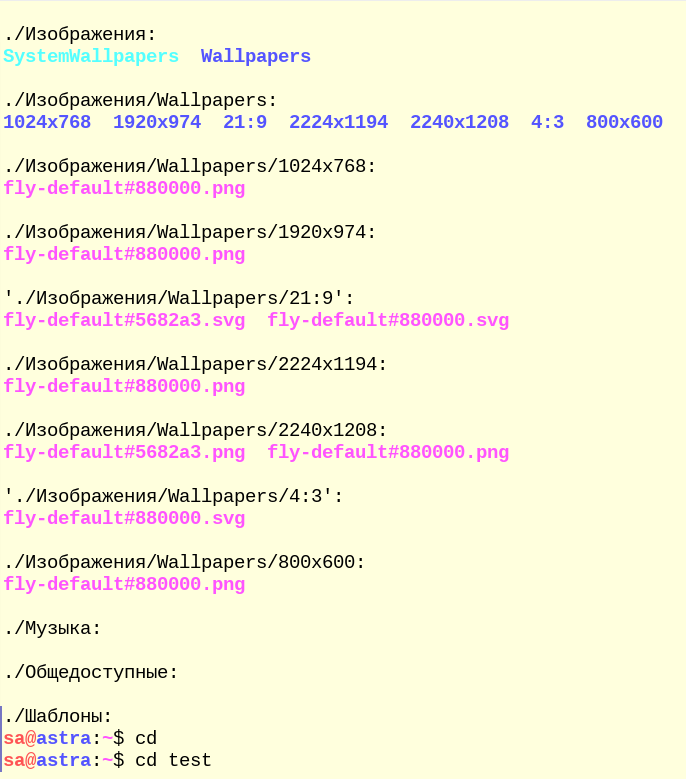


Рисунок 17 – Работа с каталогами

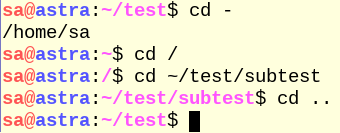


Рисунок 18 – Работа с каталогами

Список основных команд для работы с файлами. К основным командам можно отнести следующие: «touch» — создать файл или изменить временные метки у существующего файла. «realpath» — узнать абсолютный путь к файлу. «file» — определить тип файла. «stat» — получить информацию о файле (размер файла, дата создания файла и т. д.) и проверить существование файла. «cp» — копировать файл «file1» в файл с именем «file2». Если файл с таким же названием уже существует, то он будет заменён. «mv» — переименовать файл «file1» в «file2». «mv» — переместить файл в каталог. «rm» — удалить файл. «find» — найти файл. «sh» — запустить файл со сценарием Bash. «./file» — запустить исполняемый файл.

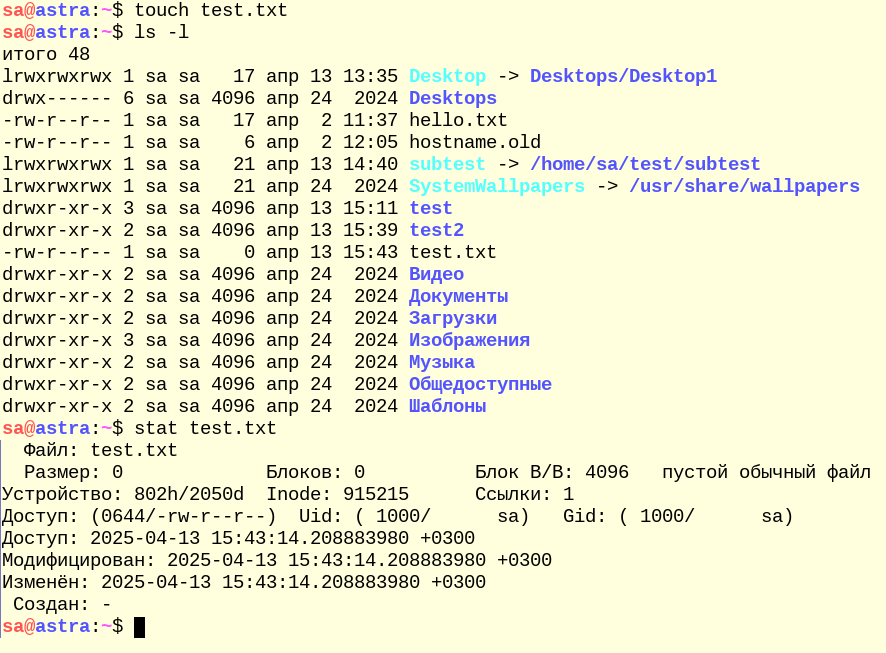


Рисунок 19 – Работа с файлами

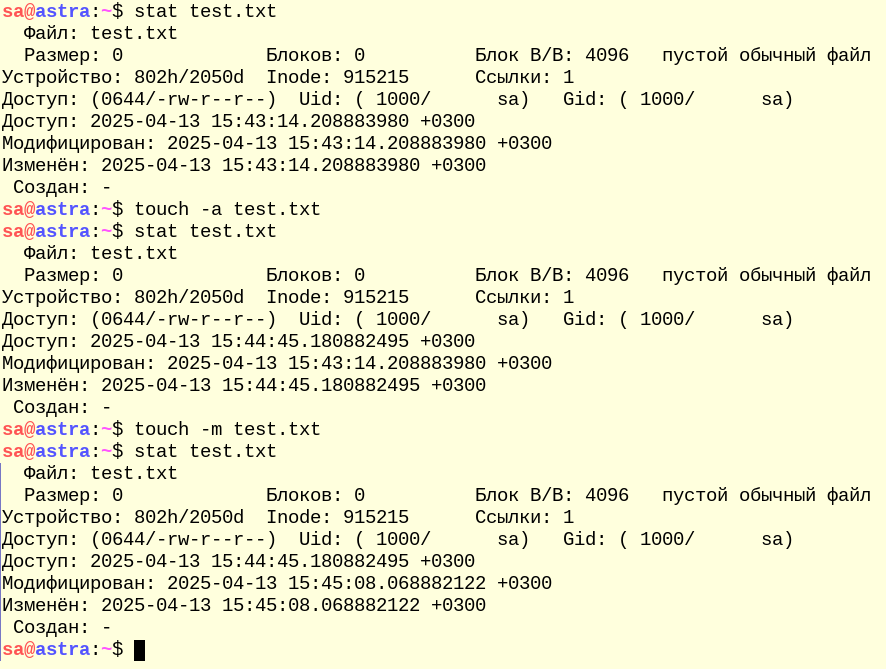


Рисунок 20 – Работа с файлами

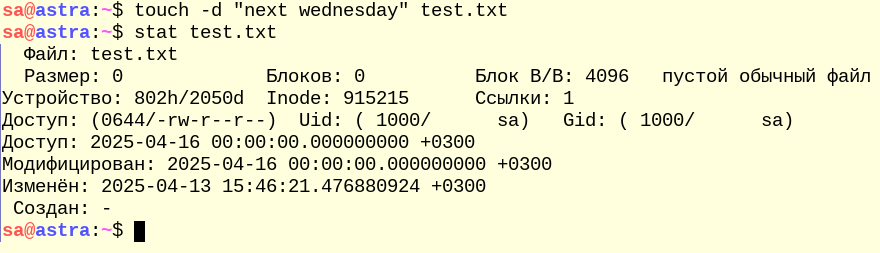


Рисунок 21 – Работа с файлами

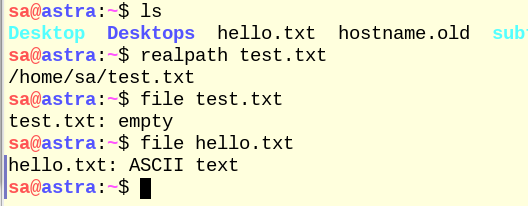


Рисунок 22 – Работа с файлами

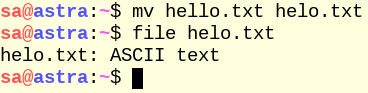


Рисунок 23 – Работа с файлами

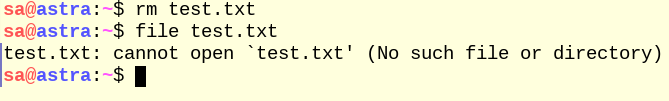


Рисунок 24 – Работа с файлами

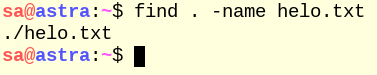
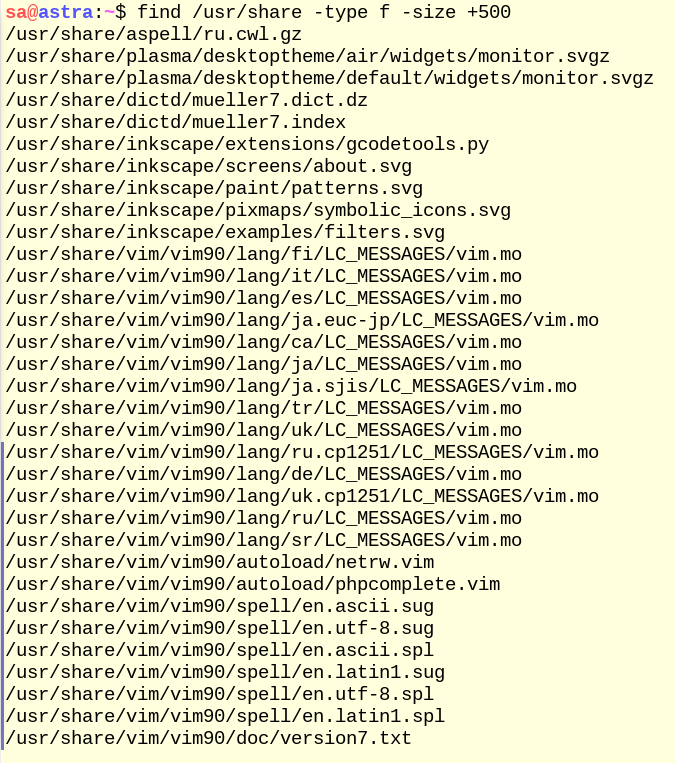


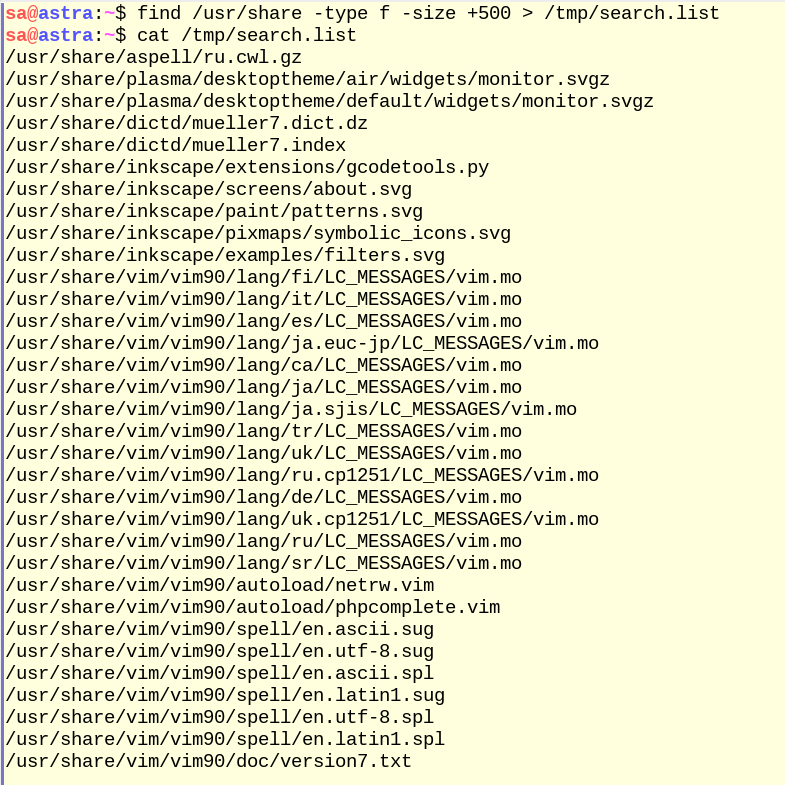
Рисунок 25 – Работа с файлами

**Задание 1**

**1. Найдите в директории /usr/share все файлы размером более 500 Кб**

****

**2. Полученный в предыдущем пункте список сохраните в файл /tmp/search.list.**



**3. Выясните размер файла /tmp/search.list в килобайтах и сохраните это значение в файл /tmp/search-size.txt.**

****

**4. Выясните количество строк в файле /tmp/search.list и запишите результат в файл /tmp/search-size.txt, сохранив предыдущие данные.**

****

**Задание 2**

**1. Создайте одной командой директорию /tmp/block6/task/1/2/3 и переместите в нее так же одной командой файлы /tmp/search.list и /tmp/search-size.txt.**



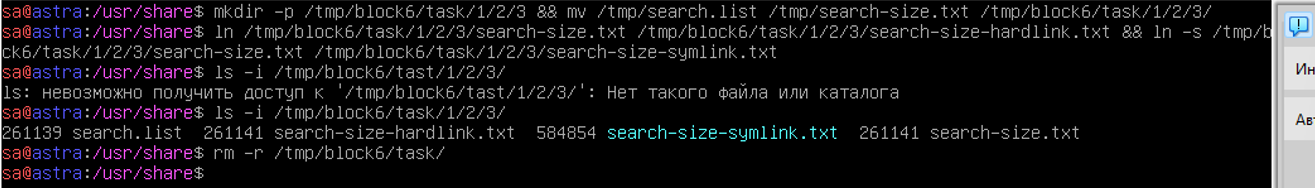
**2. Создайте жесткую и символическую ссылки на файл /tmp/block6/task/1/2/3/search-size.txt.**

****

**3. Выведите содержимое каталога /tmp/block6/task/1/2/3/ с получением айноды всех файлов.**



**4. Удалите созданный каталог /tmp/block6/task/1/2/3/ вместе с содержимым.**

****

**Вопросы**

**1. Какой каталог содержит конфигурационные файлы?**

etc

**2. Можно ли в ОС Linux создать два файла /tmp/test.txt и /tmp/TEST.txt?**

Да, можно, Файлы test.txt и TEST.txt будут считаться разными.

**3. Какая максимальная длина имени файла в ОС Linux?**

Имя не может превышать 255 байт, а общая длина пути (включая все компоненты) не может превышать 4096 байт

**4. Каким символом в выводе команды ls -l помечаются обычные файлы?**

Они помечаются символом “–“ в первом столбце вывода “ls -l”.

**5. Можно ли найти неименованный канал в структуре каталогов?**

Нет, неименованные каналы существуют только в памяти и не представлены в файловой системе как файлы или каталоги. Они используются для межпроцессного взаимодействия и исчезают после завершения процессов, использующих их. Но именованные каналы можно найти в структуре каталогов.

**6. Каким ключом команды ls можно отобразить айноду?**

Команда ls -i выведет номер айноды.

**7. Можно ли создать жесткую ссылку на файл, расположенный на другой файловой системе?**

Нет, нельзя. Жесткие ссылки могут существовать только в пределах одной файловой системы. Они указывают непосредственно на айноды файла. А последние являются уникальными для каждой файловой системы.

**8. Путь «tmp/1/2.txt» является абсолютным или относительным?**

Относительным, так как абсолютный путь всегда начинается с “/”. Абсолютный путь — это полный путь к файлу или каталогу, начинающийся с корневого каталога файловой системы. Он однозначно определяет местоположение файла или каталога независимо от текущего рабочего каталога. Относительный путь — это путь к файлу или каталогу относительно текущего рабочего каталога. Текущий рабочий каталог — это каталог, в котором вы сейчас находитесь в командной строке или в файловом менеджере.

**9. Что будет при вызове команды touch с существующим файлом?**

Данная команда обновит время последнего доступа и время последнего изменения существующего файла до текущего времени. А если же файл не существует, то создаст пустой файл.

**Вывод**

Из этой лабораторной работы я узнал, что такое стандарт FHS, почему «всё есть файл», и познакомился с необычными видами файлов в Linux.