Федеральное государственное бюджетное образовательное учреждение

высшего образования

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(МГТУ им. Н. Э. Баумана)*** |

Факультет ***Информатика и системы управления***

Кафедра ***Компьютерные системы и сети (ИУ6)***

**Отчет**

**по лабораторной работе №6**

**Дисциплина: Операционные системы**

**Название лабораторной работы: Исследование методов организации внешней памяти.**

Студент гр. ИУ6 - 54 **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_**А.А. Шумаков**\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_**В.Ю.Мельников\_**\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2018

**Цель работы**: исследование файловых систем, применяющихся в UNIX-подобных системах, а также изучение основных утилит для работы с файлами.

**Выполнение работы**

**Используемые термины**

Терминология UNIX-систем отличается от терминологии Windows. Ниже приведены термины, которые будут использованы в данной ЛР.

*Индексный дескриптор (inode)* – структура данных, хранящая метаинформацию. Каждый индексный дескриптор имеет уникальный идентификатор в файловой системе.

*Жёсткая ссылка (hard link, hardlink)* – составляющая каталога, ассоциирует имя файла с содержимым на диске. Прописывается в индексном дескрипторе файла.

*Символьная ссылка (символическая ссылка, мягкая ссылка, symbolic link, symlink)* – представляет собой структуру данных, которая содержит в себе только ссылку на другой файл. Символьные ссылки часто используются как псевдонимы. В Windows тоже существуют символьные ссылки. Доступ к символьной ссылке не ограничен, права доступа проверяются при попытке доступа к целевому файлу.

*Архиватор* – утилита для создания архивов. В отличие от Windows, где под архивом понимается файл, содержащий сжатые данные, в UNIX-подобных системах архив может быть и не сжатым, это просто информация для длительного хранения, обращение к которой происходит достаточно редко. Утилиты для сжатия файлов называются компрессорами.

Следует отметить, что все устройства в UNIX являются файлами. Все они монтируются в */dev*. Примерами могут являться */dev/sda* (первый жёсткий диск со SCSI-интерфейсом), /dev/hda (жёсткий диск с IDE-интерфейсом), /dev/null (специальное устройство, в него можно записать информацию любого объёма, при чтении выдаёт EOF, часто используется для перенаправления ненужных сообщений).

Специальными типами файлов являются также *каналы (pipes)* и *порты* (*sockets*, в некоторой литературе - *гнёзда*).

Имена папок в UNIX тоже имеют смысл. Например, */etc* – директория, содержащая конфигурационные файлы (сокращение от et cetera – «и так далее»), */bin* содержит исполняемые файлы (*binary*), */home* – домашние директории пользователей, */lib* – библиотеки (*libraries*) и т.д.

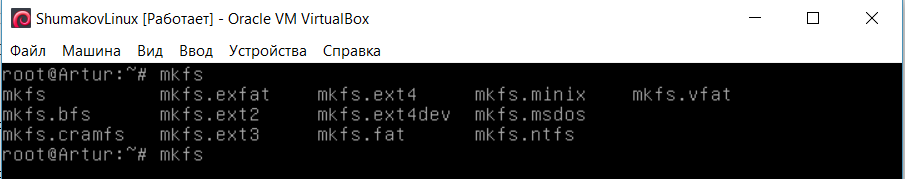
Каждая директория имеет минимум две ссылки: ‘.’ – ссылка на саму же директорию, и ‘..’ – ссылка на родительскую директорию. Эти ссылки являются жёсткими.

**Основные команды, используемые при работе с ФС**

*mount* – подключение (монтирование) файловой системы.

*umount* – отключение файловой системы.

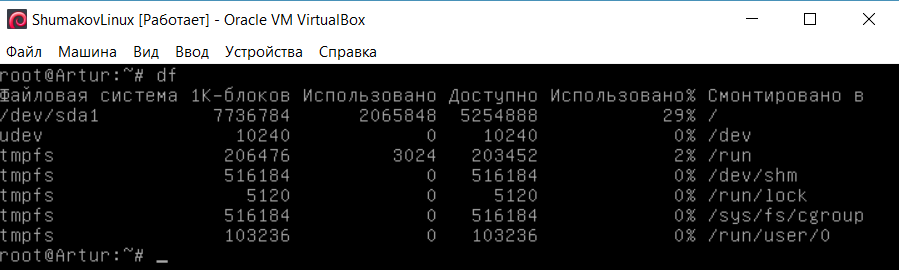
*mkfs* – создание новой файловой системы. Существуют специализированные утилиты для создания ФС определённого типа (например, *mkfs.ext4*, *mkfs.ntfs*).



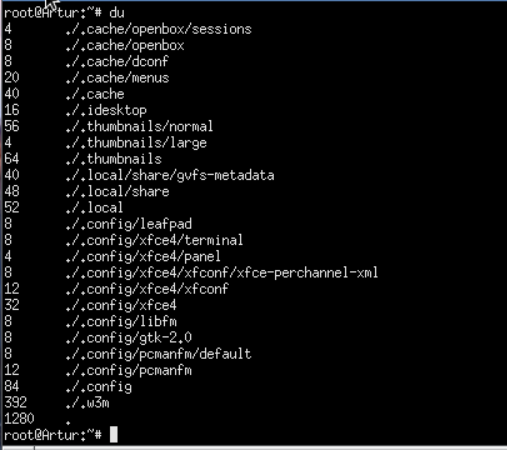
*mke2fs* – создание *ext2/3/4 ФС*.

*fdisk* – утилита для управления разделами диска.

*df* – отображает информацию об использовании диска.



*du* – отображает информацию об использовании диска.



*ls* – вывод списка директорий (аналогичные утилиты – *dir, vdir*).

*cp* – копирование файла.

*mv* – перемещение файла (альтернативное применение – переименование).

*rm* – удаление.

*cd* – смена директории.

*dd* – утилита для копирования и конвертации файлов.

*ln* – создание ссылки.

*pwd* – вывод рабочего каталога.

*mkdir* – создание папки.

*rmdir* – удалить пустую папку.

*basename* – убирает из имени файла имена каталогов и суффиксы (преобразование полного пути в имя файла).

*dirname* – преобразует полный путь к файлу в имя родительской директории.

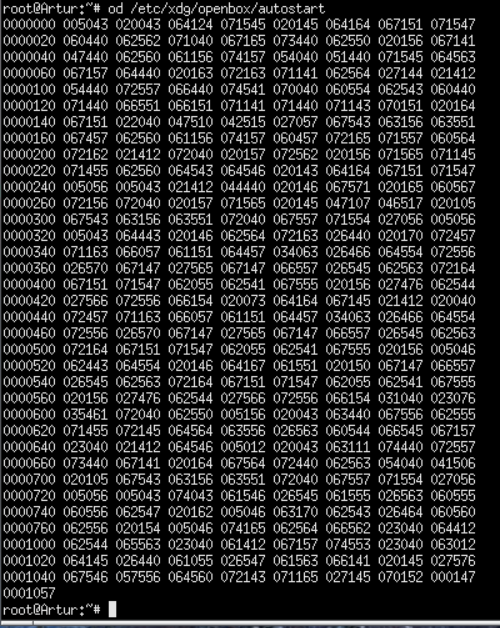
*cat* – просмотр файла.

*head* – вывод первых строк файла.

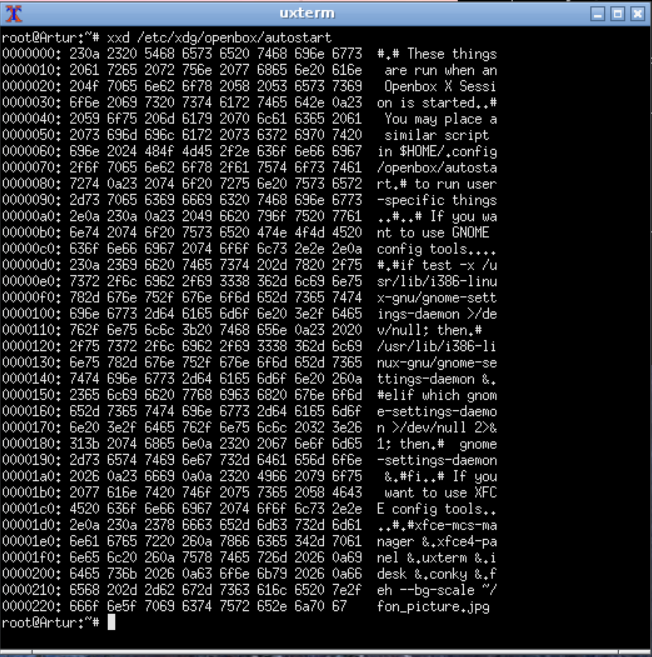
*tail* – вывод последних строк файла.

*less* – текстовый редактор.

*od* – просмотр в восьмеричном, десятичном или шестнадцатеричном формате.



*xxd* – просмотр файлов в шестнадцатеричном или двоичном формате.



*touch* – создание пустого файла.

*umask* – задание маски прав доступа для новых файлов и директорий.

*stat* – вывод атрибутов файлов и директорий.

*file* – определение типа файла.

*size* – определение размера файла.

*chown* – изменение владельца или группы для файла или директории.

*chgrp* – изменение группы файла или директории. Может быть исполнена обычным пользователем, но в таком случае могут быть изменены только те группы, членами которых является пользователь.

*chmod* – смена прав доступа.

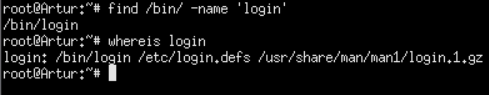
*chattr* – смена атрибутов файла в ext2.

*cksum* – расчет количества байт и контрольной суммы файла.

*fuser* – идентификация процессов, использующих указанные файлы или порты.

*find* – утилита поиска.

*whereis* – поиск файлов определённого приложения.



*which* – отображает полный путь к программе или скрипту.



*cmp* – утилита сравнения, может быть использована для файлов.

*diff* – утилита сравнения файлов, выводящая разницу. Утилита diff3 сравнивает 3 файла.

*cpio* – архиватор.

*tar* – компрессор tar.

*xz* – компрессор xz.



*bzip2* – компрессор.

*gzip* – компрессор.

*mkswap* – создание файла подкачки (swap).

*mktemp* – создание временного файла.

**Файловая система ext4**

Некоторые особенности ext4fs:

* Экстенты (extents) – вместо поблочной адресации допустимо адресовать до 128 Мб идущих подряд блоков одним дескриптором.
* Контрольная сумма в журнале – в отличие от ext2, ext3 и ext4 являются журналируемыми файловыми системами, в ext4 хранится контрольная сумма транзакций диска.
* Мультиблочное выделение (выделение блоков группами, multiblock allocation) – файловая система хранит данные не только об отдельных свободных блоках, но и о группах свободных блоков. При записи файла производится поиск группы блоков, на которые данные могут быть записаны без фрагментации.
* Отложенное выделение (delayed allocation) – выделение блоков происходит перед физической записью на диск (а не при системном вызове write).
* Предварительное выделение (persistent pre-allocation) – возможно заранее зарезервировать место для записи файла.
* Версия inode (индексного дескриптора) – у индексного дескриптора появился номер, увеличивающийся каждый раз при изменении дескриптора.
* Обратная совместимость – файловые системы ext2 и ext3 могут быть монтированы как ext4.

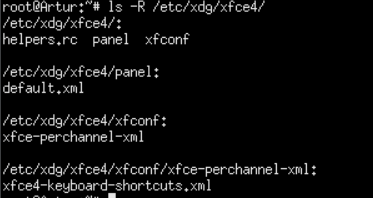
Файловая система ext4 используется по умолчанию в дистрибутив Debian 7.2

В качестве практики будет проведено исследование возможностей утилит *ls* и *ln*.

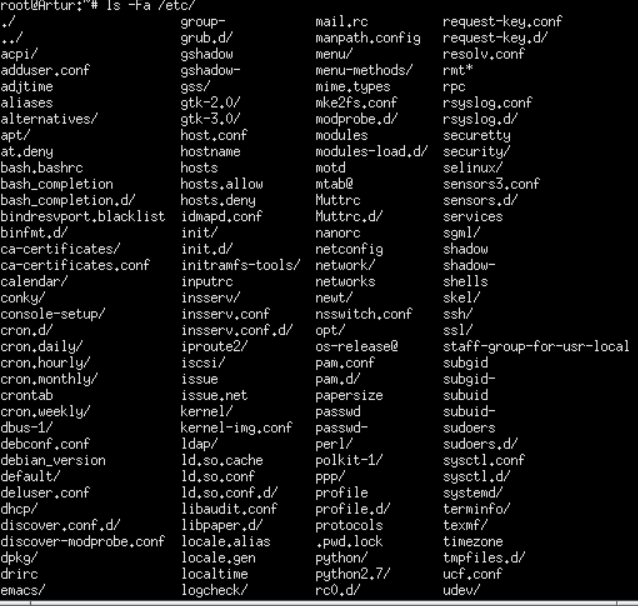
Некоторые ключи, использующиеся с утилитой *ls*:

-F – добавлять ‘/’ для каждого каталога, ‘|’ для каналов, ‘\*’ для исполняемых файлов.

-R – рекурсивная выдача каталогов.



-a – включает в список файлы, начинающиеся с ‘.’



-i – показывать номера inode (индексных дескрипторов).

-l – выдавать следующие данные: тип файла, права доступа, количество ссылок, имя владельца, имя группы, размер (в байтах) файла, штамп времени (временной штамп, timestamp) и имя файла. Кодирование типов файлов:

- – регулярный (обычный) файл;

d – каталог;

b – блочное устройство;

c – символьное устройство;

p – канал (pipe);

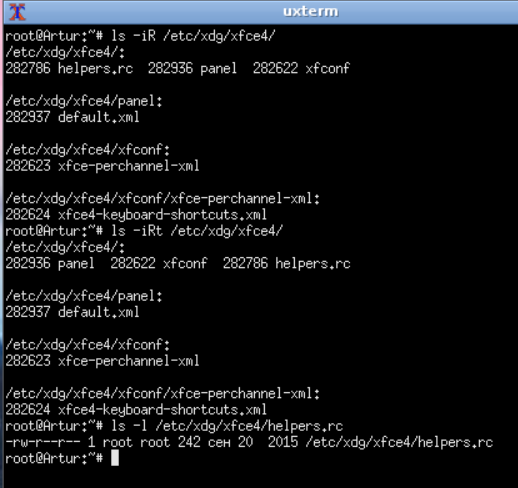
l – символьная ссылка;

s – порт (socket).

-d – обрабатывать каталог как файл.

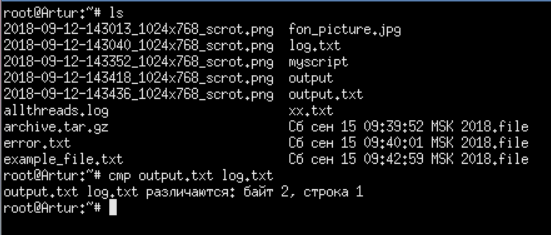
-r – сортировка в обратном порядке.

-t – сортировать по штампу времени.

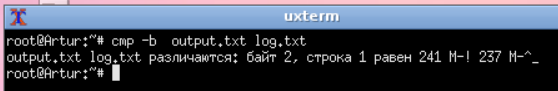


В качестве дополнительного задания покажим работу (с различными опциями) следующих утилит: cmp, diff, fuser.

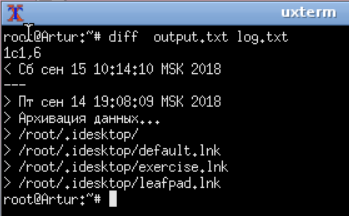
*cmp output.txt log.txt*



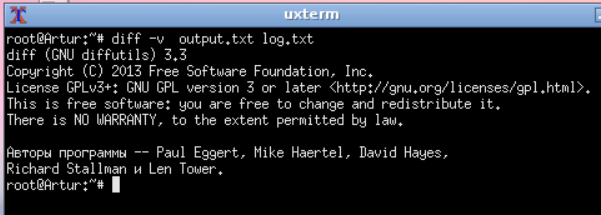
*cmp –b output.txt log.txt*



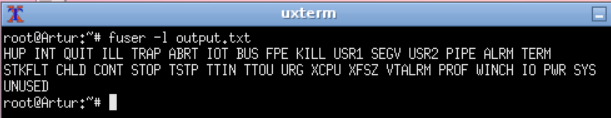
*diff output.txt log.txt*



*diff –v output.txt log.txt*



*Fuser –l output.txt*



Утилита *ln* позволяет создавать как жёсткие, так и символьные ссылки. Создание жёсткой ссылки на файл:

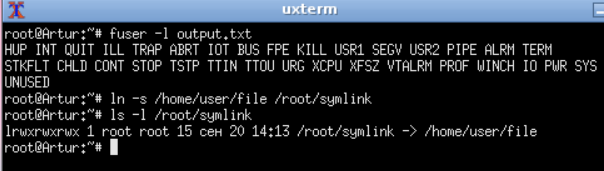
*ln file hardlink*

С утилитой *ln* используются следующие ключи:

-s – создать символьную ссылку.

-d – сделать жёсткую ссылку для директории (только для суперпользователя).

-b – делать резервные копии файлов, которые будут удалены или изменены.



Также значения ссылок можно считывать с помощью утилиты *readlink:*

*readlink /root/symlink*

**Файловая система NFS**

NFS (Network File System), по сути, представляет собой не самостоятельную файловую систему, а протокол для сетевого доступа к файловым системам. NFS абстрагируется от файловых систем сервера и клиента, можно представить, что NFS – это файловая система над файловыми системами.

Сервер NFS экспортирует свои каталоги, которые будут использоваться клиентами (обычно их список хранится в /etc/exports). Клиенты монтируют эти каталоги, монтируемые каталоги становятся частью иерархии файловой системы клиентов, хотя на самом деле они остаются на сервере и доступ к ним осуществляется с помощью механизма удалённого вызова процедур.

NFS в Linux реализована с помощью нескольких уровней. Верхний уровень – уровень системных вызовов, он алогичен таковому для локальных ФС (обработка системных вызовов открытия файла, записи и т.д.). После обработки системный вызов передаётся на уровень виртуальной файловой системы (VFS). Виртуальная файловая система обслуживает таблицы, содержащие записи виртуальных индексных дескрипторов (virtual inode, v-узел) для всех открытых файлов. Нижний уровень – локальные файловые системы.

**Файловая система GPFS**

GPFS (General Parallel File System) – параллельная кластерная файловая система от IBM, впервые представлена IBM в 1998 году. Используется в кластерах (часто в IBM x), при организации облачных инфраструктур, совместно с суперкомпьютерами (IBM Blue Gene).

Файл в GPFS может быть распределён по нескольким дискам, находящимся на различных узлах, при этом для работы с файловой системой могут использоваться обычные утилиты UNIX, как в случае с локальной ФС.

Преимущества GPFS:

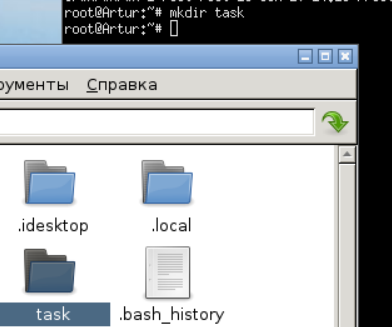
* Высокая производительность – ресурсы GPFS доступны одновременно многим процессорам на многих узлах.
* Высокий показатель восстанавливаемости и высокая доступность – GPFS является журналируемой файловой системой, причём для каждого узла ведётся собственный журнал. GPFS включает в себя утилиты для гибкой настройки и администрирования.
* Гибкость – узлы могут добавляться и удаляться без перемонтирования файловой системы.

IBM также отмечает, что, в отличие от NFS, GPFS является самостоятельной файловой системой, что положительно сказывается на быстродействии и удобстве администрирования.

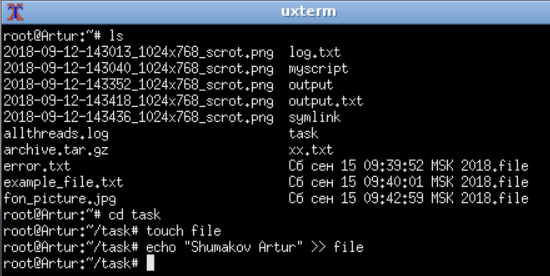
**Выполнение заданий лабораторной работы**

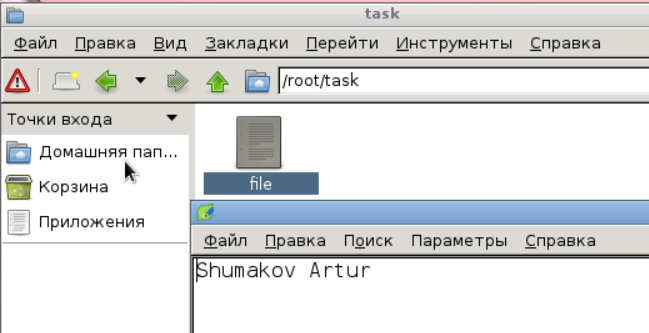
* **Создать в домашнем каталоге папку**

Команда для создания папки в домашнем каталоге*: mkdir*



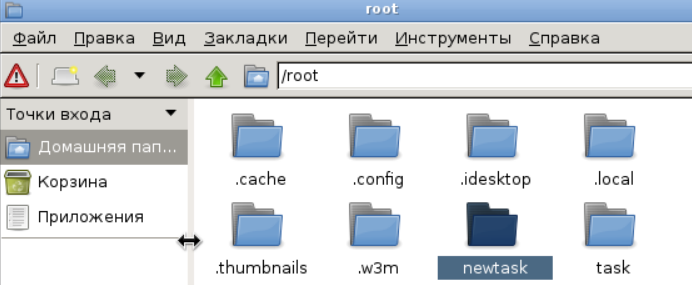
* **Создать в папке файл и записать в него текст**



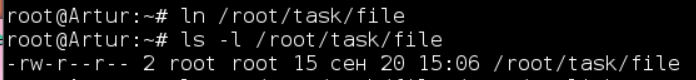


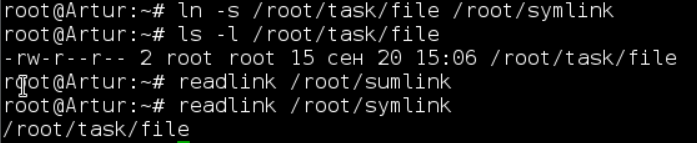
* **Скопировать папку вместе с файлом (одной командой)**





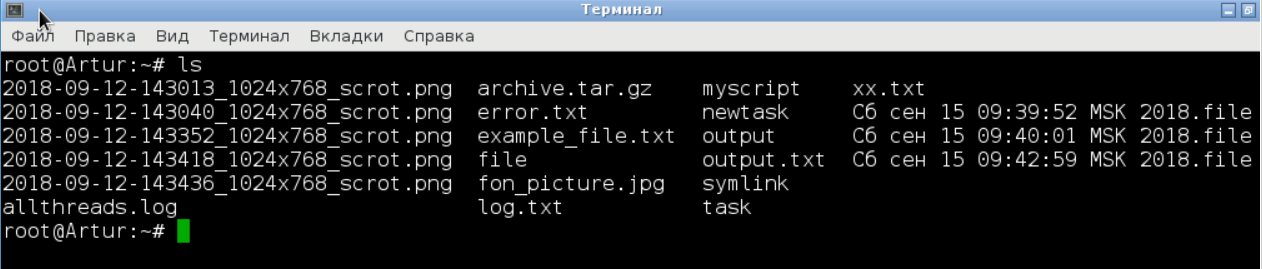
* **Создать в домашнем каталоге жёсткую и мягкую ссылки на ваш файл**



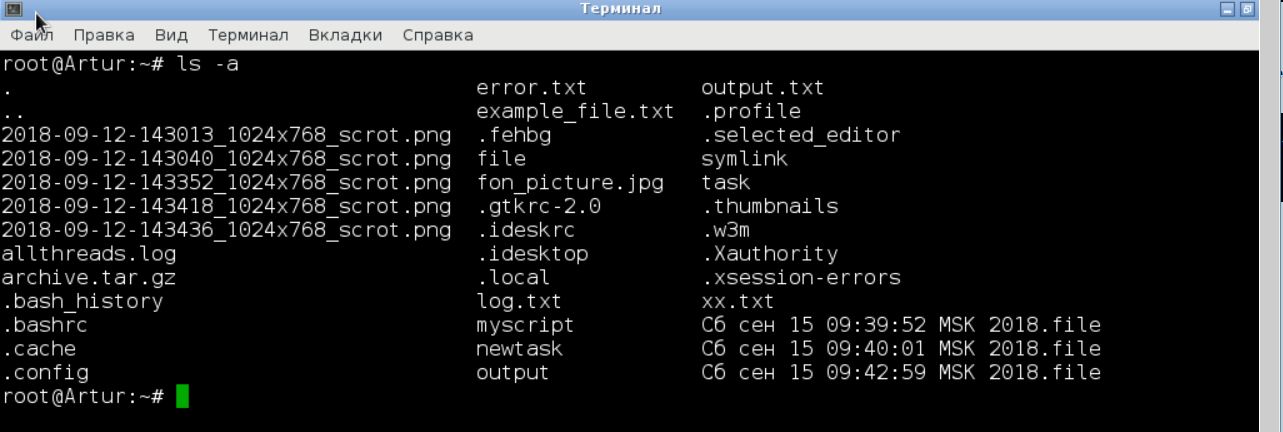


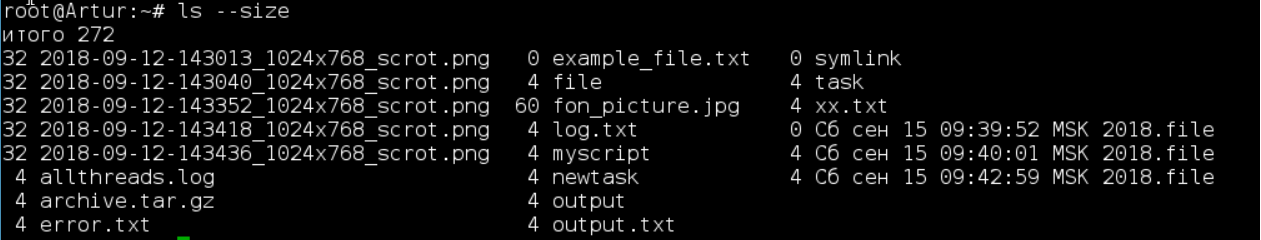
Как видно из рисунка, ls не учитывает символьные ссылки, потому что они хранятся как отдельные структуры, в отличие от жёстких ссылок.

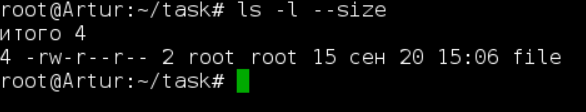
* **Вывести в консоль содержимое домашней папки**



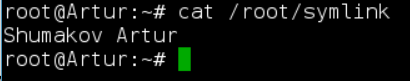
* **Вывести в консоль содержимое домашнего каталога, включая скрытые файлы и папки в широком формате. Определить размер вашего файла.**





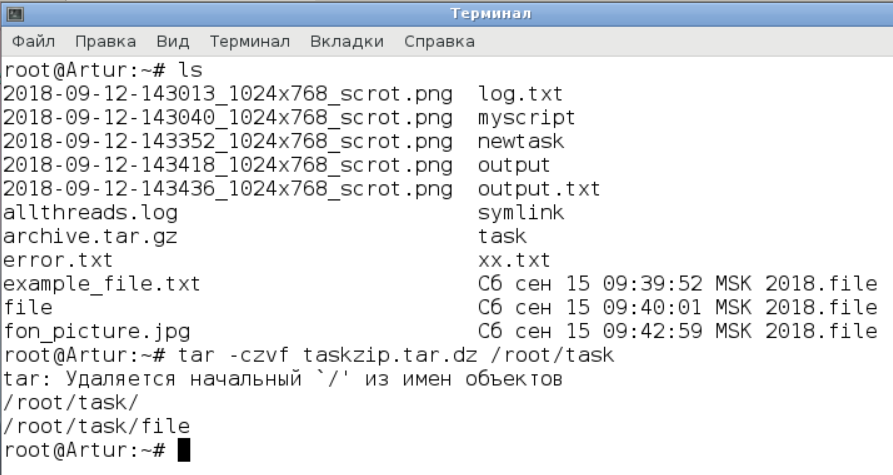


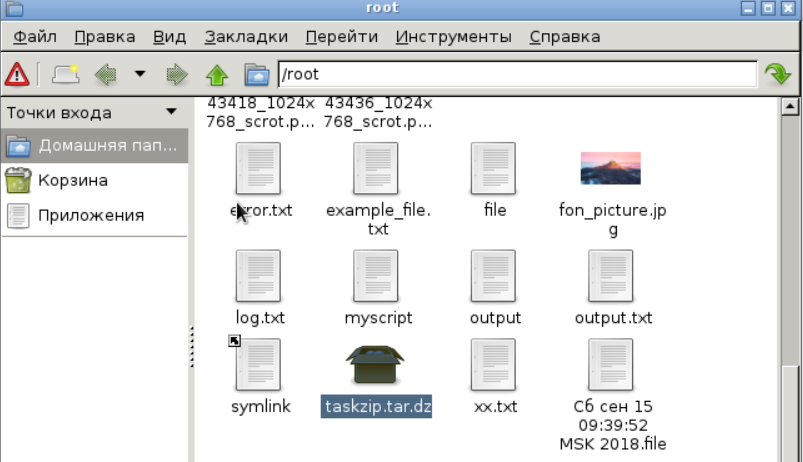
* **Вывести текст вашего файла в консоль, используя ссылку**



* **Создать архив (gz) вашей папки**

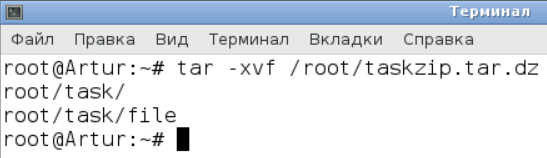
|  |  |  |
| --- | --- | --- |
| **с** | «create» | создать файл архива |
| **v** | «verbose» | выводить информацию в процессе выполнения |
| **f** | «file» | использовать имя файла архива указанное после ключей. Если не указать ключ f то команда будет использовать настройки по умолчанию либо выведет результат прямо в консоль |
| **z** | «gzip» | запаковать файл при помощи gzip |

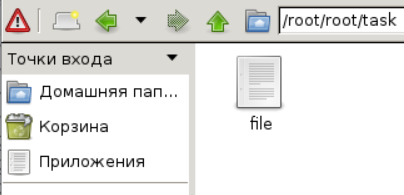




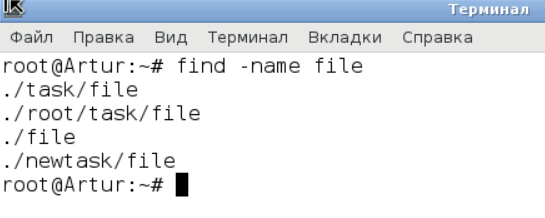
* **Создать новую папку и распаковать в неё архив**

|  |  |  |
| --- | --- | --- |
| **x** | «eXtract» | извлечь файлы |
| **v** | «verbose» | выводить информацию в процессе выполнения |
| **f** | «file» | использовать имя файла архива для распаковки указанное после ключей |

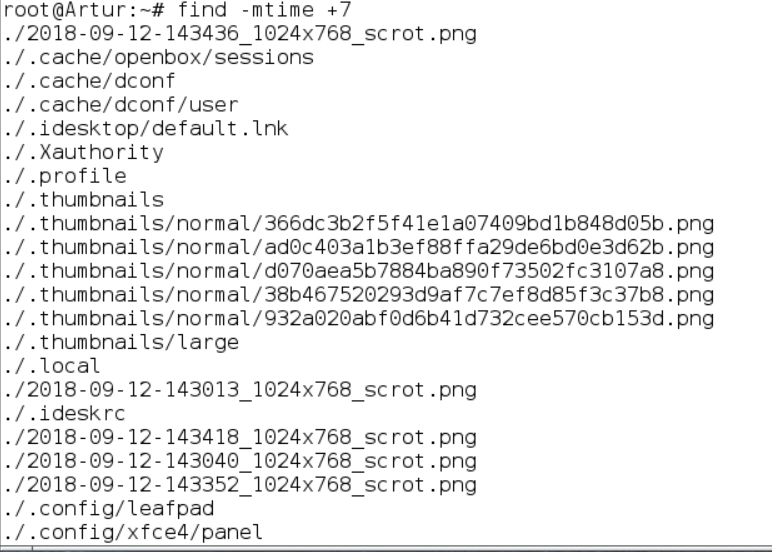




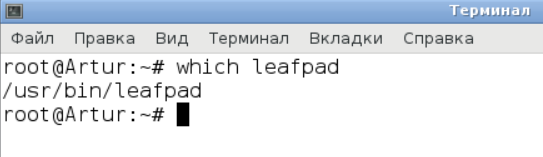
* **Найти в вашем домашнем каталоге и его подкаталогах файл с заданным именем**



* **Найти в вашем домашнем каталоге и его подкаталогах файлы с датой создания старше 7 дней от текущей даты.**



* **Определить, из какого каталога запускается leafpad**



**Вывод:** В данной ЛР были рассмотрены некоторые из применяющихся на данный момент в UNIX-подобных системах файловые системы. Также были рассмотрены основные утилиты для работы с файлами.

Файловая система ext4 представляет собой развитие файловой системы ext3. Она может содержать большее количество каталогов, адресовать б*о*льшие диски, эффективнее использует пространство при размещении больших файлов. ФС ext4 используется по умолчанию в ряде дистрибутивов, в том числе и Debian (начиная с версии 7.0).

Файловая система NFS основана на удалённом вызове процедур (RPC) и предоставляет протокол для взаимодействия с удалёнными ресурсами. NFS не является самостоятельной ФС, она позволяет монтировать папки из различных файловых систем.

GPFS является высокопроизводительной параллельной файловой системой. В отличие от NFS, GPFS является самостоятельной ФС. Эта ФС является проприетарной. Она может использоваться как в системах высокой доступности, так и в высокопроизводительных системах.