

ЛАБОРАТОРНАЯ РАБОТА №3

Изучение файловой системы ОС Linux

1. Основные сведения о файловой системе Linux. Для пользователя каждый файл – это отдельный предмет, у которого есть начало и конец и который отличается от всех остальных файлов именем и расположением ("как называется" и "где лежит"). Как и любой предмет, файл можно создать, переместить и уничтожить, однако без внешнего вмешательства он будет сохраняться неизменным неопределенно долгое время. Файл предназначен для хранения данных любого типа – текстовых, графических, звуковых, исполняемых программ и многого другого. Аналогия файла с предметом позволяет пользователю быстро освоиться при работе с данными в операционной системе.

Для операционной системы Linux файл – не менее важное понятие, чем для ее пользователя: все данные, хранящиеся на любых носителях, обязательно находятся внутри какого-нибудь файла, в противном случае они просто недоступны ни для операционной системы, ни для пользователей. Более того, многие устройства, подключенные к компьютеру (начиная с клавиатуры и заканчивая любыми внешними устройствами, например, принтерами и сканерами), Linux представляет как файлы (так называемые файлы-дырки). Конечно, файл, содержащий обычные данные, сильно отличается от файла, предназначенного для обращения к устройству, поэтому в Linux определено несколько различных типов файлов. В основном пользователь имеет дело с файлами трех типов: *обычными файлами*, предназначенными для хранения данных, *каталогами* и *файлами-ссылками*.

Файл – это отдельная область данных на одном из носителей информации, у которой есть собственное имя.

С точки зрения пользователя файловая система – это «пространство», в котором размещаются файлы. Наличие файловой системы позволяет определить не только "как называется файл", но и "где он находится". Различать файлы только по имени было бы нецелесообразно: приходилось бы помнить, как называется каждый файл и при этом заботиться о том, чтобы имена никогда не повторялись. Более того, необходим механизм, позволяющий работать с группами тематически связанных между собой файлов. Иначе говоря, файлы нужно систематизировать.

Файловая система – способ хранения и организации доступа к данным на информационном носителе или его разделе. Классическая файловая система имеет иерархическую структуру, в которой файл однозначно определяется полным путем к нему. Linux может работать с различными типами файловых систем, которые различаются списком поддерживаемых возможностей, производительностью в разных ситуациях, надежностью и другими признаками. В последнее время Linux использует такие файловые системы как *ext2*, *ext3* и *ext4*.

Большинство современных файловых систем используют в качестве основного организационного принципа каталоги. *Каталог* – это список ссылок на файлы или другие каталоги. Принято говорить, что каталог содержит файлы или другие каталоги, хотя в действительности он только ссылается на них, физическое размещение данных на диске обычно никак не связано с размещением каталога. Каталог, на который есть ссылка в данном каталоге, называется подкаталогом или вложенным каталогом. Каталог в файловой системе более всего напоминает библиотечный каталог, содержащий ссылки на объединенные по каким-то признакам книги и другие разделы каталога (файлы и подкаталоги). Ссылка на один и тот же файл может содержаться в нескольких каталогах одновременно - это делает доступ к файлу более удобным. В файловой системе *ext2* каждый каталог - это отдельный файл особого типа ("d", от

англ. "directory"), отличающийся от обычного файла с данными: в нем могут содержаться только ссылки на другие файлы и каталоги.

В файловой системе Linux возможности каталогов и файлов гораздо шире, чем в ОС Windows.

Довольно часто вместо термина "каталог" употребляется "папка" (англ. folder). Этот термин хорошо вписывается в представление о файлах как о предметах, которые можно раскладывать по папкам, однако часть возможностей файловой системы, которая противоречит этому представлению, таким образом затемняется. В частности, с термином "папка" плохо согласуется то, что ссылка на файл может присутствовать одновременно в нескольких каталогах, файл может быть ссылкой на другой файл и т. д. В Linux эти возможности файловой системы весьма важны для эффективной работы, поэтому мы будем использовать более подходящий термин "каталог".

В файловой системе, организованной при помощи каталогов, на любой файл должна быть ссылка как минимум из одного каталога, в противном случае файл просто не будет доступен внутри этой файловой системы, иначе говоря, не будет существовать. Таким образом, ссылаясь на один и тот же файл из разных каталогов достигается определенное удобство. В то же время нет необходимости копировать данный файл во все эти каталоги, занимая свободное место диска.

Файлом в ОС Linux называется всё: жесткий диск, раздел на жестком диске, параллельный порт, подключение к веб-сайту, карты Ethernet и каталоги. Для Linux файл – это просто поток байтов, а тип вовсе не определяет его содержимое (в отличие от ОС Windows). Разделение файлов согласно их содержанию оставляется приложениям.

Файл можно найти по его имени. В ОС Linux содержимое файла связывается с его именем при помощи *жестких ссылок*. Создание файла любым способом означает, что будет создана жесткая ссылка – имя файла, и открыта новая область данных на диске. Причем количество ссылок на одну и ту же область данных может быть любым, т.е. у файла может быть несколько имен. Пользователь может добавлять ссылки на файл самостоятельно.

Доступ к одному и тому же файлу по разным ссылкам может потребоваться в следующих случаях:

- Одна и та же программа известна под несколькими именами;
- Доступ пользователей к некоторым каталогам может быть ограничен из соображений безопасности. Однако если все же нужно организовать доступ пользователей к файлу, который находится в таком каталоге, можно создать жесткую ссылку на этот файл в другом каталоге;
- При большом количестве вложенных друг в друга каталогов полные имена файлов становятся слишком длинными. С такими именами трудно обращаться пользователю. Можно создать новые ссылки на такие файлы в более доступном каталоге.

В файловых системах Linux вся информация, необходимая для работы с файлом, хранится в *индексном дескрипторе (inode)*. Каждому файлу и каталогу соответствует свой собственный индексный дескриптор. Индексный дескриптор содержит следующие данные:

- тип файла (обычный файл, каталог, файл-дырка и т.д.);
- права доступа к файлу;
- информация о том, кому принадлежит файл;
- отметки о времени создания, модификации, последнего доступа к файлу;
- размер файла;
- указатели на физические блоки на диске, принадлежащие этому файлу – в этих блоках хранится содержимое файла.

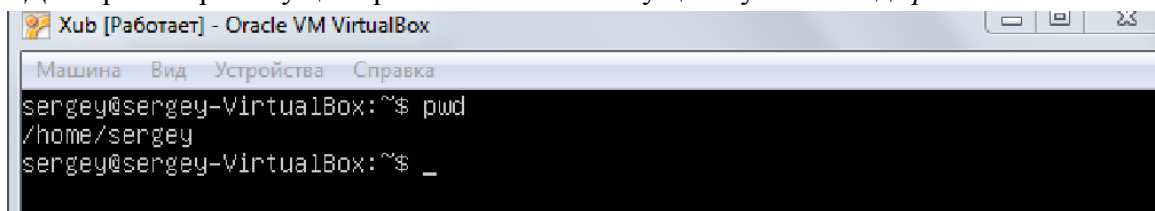
Все индексные дескрипторы пронумерованы. Поэтому индексный дескриптор – это уникальный идентификатор файла в файловой системе. Все операции с файловой системой на самом деле производятся над индексными дескрипторами. Таким образом, жесткая ссылка – это запись вида «*имя файла + номер индексного дескриптора*».

Существуют так называемые *символьные ссылки*. Символьная ссылка – это файл особого типа, в котором содержится путь к другому файлу. Если на пути к файлу встречается символьная ссылка, то система выполняет подстановку: исходный путь заменяется тем, что содержится в ссылке. Символьная ссылка может содержать путь к несуществующему файлу. В этом случае она не будет работать.

2. *Работа с файлами и каталогами в Linux*. Для работы с файлами и каталогами в Linux предусмотрен ряд команд. Будем работать в консоли. Рассмотрим некоторые из них.

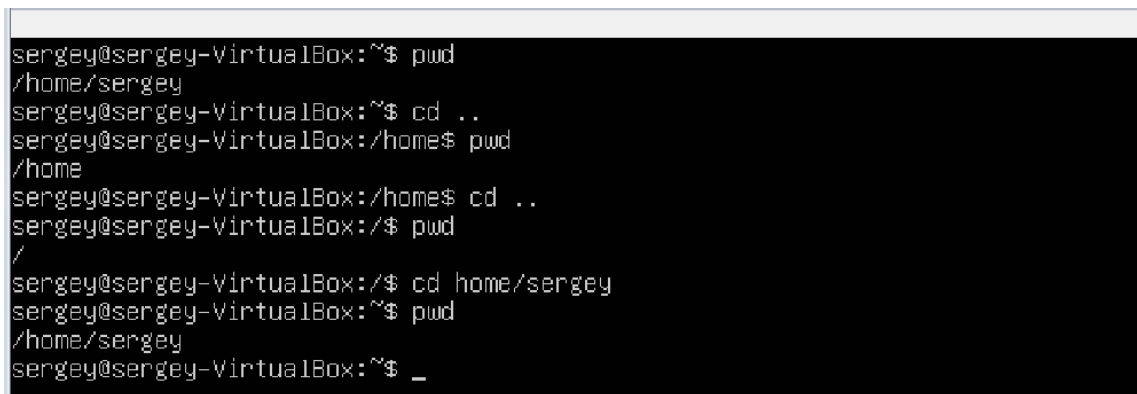
Авторизуемся в консоли №1.

Для просмотра текущего рабочего каталога существует команда *pwd*:



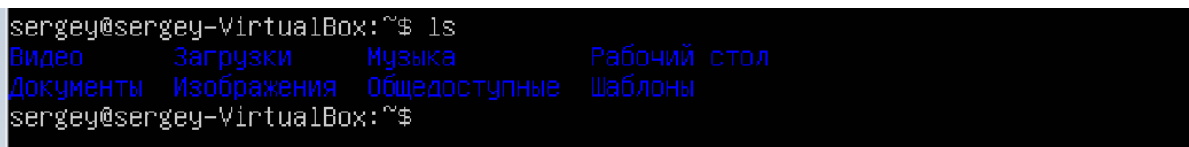
```
Xub [Работает] - Oracle VM VirtualBox
Машина Вид Устройства Справка
sergey@sergey-VirtualBox:~$ pwd
/home/sergey
sergey@sergey-VirtualBox:~$ _
```

Корневой каталог в ОС семейства Linux один и называется он «/». Для перехода в другой каталог существует команда *cd <каталог>*. Если необходимо подняться на один уровень вверх, то потребуется команда «*cd ..*» (после буквы «*d*» необходимо поставить пробел). Перейдем сначала в каталог «/», потом в каталог «*home*», а потом в каталог, который называется как и ваша учетная запись:



```
sergey@sergey-VirtualBox:~$ pwd
/home/sergey
sergey@sergey-VirtualBox:~$ cd ..
sergey@sergey-VirtualBox:/home$ pwd
/home
sergey@sergey-VirtualBox:/home$ cd ..
sergey@sergey-VirtualBox:/$ pwd
/
sergey@sergey-VirtualBox:/$ cd home/sergey
sergey@sergey-VirtualBox:~$ pwd
/home/sergey
sergey@sergey-VirtualBox:~$ _
```

Для просмотра содержимого каталога необходимо выполнить команду *ls*:



```
sergey@sergey-VirtualBox:~$ ls
Видео      Загрузки  Музыка    Рабочий стол
Документы  Изображения  Общедоступные  Шаблоны
sergey@sergey-VirtualBox:~$
```

Запустив команду *ls -a* получим скрытые файлы и директории:

```
sergey@sergey-VirtualBox:~$ ls -la
.          .config      .thumbnails  Видео          Рабочий стол
..         .dmrc        .Xauthority  Документы      Шаблоны
.bash_history .gconf       .Xdefaults   Загрузки
.bash_logout .ICEauthority .xscreensaver Изображения
.bashrc      .local       .xsession-errors Музыка
.cache       .profile     .xsession-errors.old Общедоступные
```

Команда `ls` имеет и другие модификаторы. Ознакомиться можно с ними, набрав команду `man ls`:

```
LS(1)                                User Commands                                LS(1)

NAME
  ls - list directory contents

SYNOPSIS
  ls [OPTION]... [FILE]...

DESCRIPTION
  List information about the FILES (the current directory by default).
  Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
  fied.

  Mandatory arguments to long options are mandatory for short options
  too.

  -a, --all
      do not ignore entries starting with .

  -A, --almost-all
      do not list implied . and ..

  --author
      with -l, print the author of each file

  -b, --escape
      print C-style escapes for nongraphic characters

  --block-size=SIZE

Manual page ls(1) line 1 (press h for help or q to quit)
```

Создать каталог можно командой `mkdir <имя каталога>`:

```
sergey@sergey-VirtualBox:~$ mkdir new
sergey@sergey-VirtualBox:~$ ls
file.txt  Видео  Загрузки  Музыка  Рабочий стол
new       Документы  Изображения  Общедоступные  Шаблоны
```

Возможно создать два каталога одновременно командой `mkdir <имя каталога 1> <имя каталога 2>`.

Создать пустой файл (неважно с каким расширением) в Linux можно различными способами. Например, можно воспользоваться командой `touch <имя файла>`:

```
sergey@sergey-VirtualBox:~$ touch file.txt
sergey@sergey-VirtualBox:~$ ls
file.txt  new1  Видео  Загрузки  Музыка  Рабочий стол
new       new2  Документы  Изображения  Общедоступные  Шаблоны
```

Будет создан файл `file.txt` в домашнем каталоге. Для просмотра его содержимого воспользуемся командой `cat <имя файла>`. Так как файл `file.txt` пустой, то в консоли ничего не отобразится:

```
sergey@sergey-VirtualBox:~$ cat file.txt
sergey@sergey-VirtualBox:~$
```

Команда `rm <имя файла>` позволяет удалять файлы. Удалим файл `file.txt`, созданный ранее:

```
sergey@sergey-VirtualBox:~$ rm file.txt
sergey@sergey-VirtualBox:~$ ls
new  new2  Документы  Изображения  Общедоступные  Шаблоны
new1 Видео  Загрузки  Музыка      Рабочий стол
```

Для полного удаления файла с диска необходимо удалить все жесткие ссылки на него!

Для удаления каталога необходимо использовать ту же команду, но с модификатором: *rm -r*

```
sergey@sergey-VirtualBox:~$ rm -r new1
sergey@sergey-VirtualBox:~$ rm -r new2
sergey@sergey-VirtualBox:~$ ls
new  Документы  Изображения  Общедоступные  Шаблоны
Видео  Загрузки  Музыка      Рабочий стол
```

r <имя каталога>. Удалим каталоги *new1* и *new2*:

Справку по команде *rm* можно получить, набрав команду *man rm*.

Создадим в домашнем каталоге каталог *newdir* и скопируем в него каталог *new* командой *cp -r new newdir*:

```
sergey@sergey-VirtualBox:~$ cp -r new newdir
sergey@sergey-VirtualBox:~$ ls newdir
new
```

Скопируем файл *file.txt* из домашнего каталога в каталог *new*:

```
sergey@sergey-VirtualBox:~$ pwd
/home/sergey
sergey@sergey-VirtualBox:~$ cp file.txt new/file.txt
sergey@sergey-VirtualBox:~$ ls new
file.txt  text
```

Для перемещения файлов или каталогов существует команда *mv <имя файла 1 (директории 1)> <имя файла 2 (директории 2)>*. В качестве примера переместим файл *file.txt* из домашнего каталога в каталог *newdir*:

```
sergey@sergey-VirtualBox:~$ ls
file.txt  newdir  Документы  Изображения  Общедоступные  Шаблоны
new       Видео  Загрузки  Музыка      Рабочий стол
sergey@sergey-VirtualBox:~$ mv file.txt newdir
sergey@sergey-VirtualBox:~$ ls
new  Видео  Загрузки  Музыка      Рабочий стол
newdir  Документы  Изображения  Общедоступные  Шаблоны
sergey@sergey-VirtualBox:~$ ls newdir
file.txt  new
sergey@sergey-VirtualBox:~$ _
```

Иногда полезным бывает узнать тип содержимого файла. Это можно сделать по расширению файла. Но мы знаем, что в Linux не всегда расширение определяет содержимое файла, поэтому воспользуемся командой *file <имя файла>*:

```
sergey@sergey-VirtualBox:~/new$ file file.txt
file.txt: empty
```

Файл *file.txt* пустой (empty). А вот следующая команда сообщает, что *new* - это каталог:

```
sergey@sergey-VirtualBox:~$ file new
new: directory
```

Попробуем создать второе имя (вторую жесткую ссылку) файлу *file.txt*. Наберем команду *ln <file.txt> <file_hard_link>*. По умолчанию ссылка будет создана в текущем каталоге:

```
sergey@sergey-VirtualBox:~$ ln file.txt file_hard_link
sergey@sergey-VirtualBox:~$ ls
file_hard_link  new  Видео  Загрузки  Музыка      Рабочий стол
file.txt       newdir  Документы  Изображения  Общедоступные  Шаблоны
```

Узнаем номера индексных дескрипторов (inode) всех файлов и каталогов в домашнем каталоге. Для этого наберем команду *ls -li*:

```
sergey@sergey-VirtualBox:~$ ls -l
438742 file_hard_link 438135 Видео 438133 Музыка
438742 file.txt 438100 Документы 438066 Общедоступные
438645 new 438064 Загрузки 438063 Рабочий стол
438658 newdir 438134 Изображения 438065 Шаблоны
```

Видно, что у файлов *file.txt* и *file_hard_link* одинаковый индексный дескриптор. Этого говорит о том, что у файла *file.txt* два имени. Аналогичным образом можно добавить любое количество имен. При этом (а так обычно и бывает) ссылки находятся в разных каталогах.

Создадим теперь *символьную ссылку* на файл *file.txt*. Воспользуемся командой *ln -s <file.txt> <file_sym_link>*. Ссылка будет создана в текущем каталоге.

Для просмотра каталога мы воспользовались командой *ls* с ключом «*-li*». Видим, что индексный дескриптор уже другой (438669), но далее следует символ «*->*», который сообщает, что *file_sym_link* является указателем на файл *file.txt*:

```
sergey@sergey-VirtualBox:~$ ln -s file.txt file_sym_link
sergey@sergey-VirtualBox:~$ ls -li
total 40
438742 -rw-rw-r-- 2 sergey sergey 0 авг. 29 19:24 file_hard_link
438669 lrwxrwxrwx 1 sergey sergey 8 авг. 29 19:49 file_sym_link -> file.txt
438742 -rw-rw-r-- 2 sergey sergey 0 авг. 29 19:24 file.txt
438645 drwxrwxr-x 2 sergey sergey 4096 авг. 28 22:00 new
438658 drwxrwxr-x 3 sergey sergey 4096 авг. 28 22:05 newdir
438135 drwxr-xr-x 2 sergey sergey 4096 авг. 25 13:03 Видео
438100 drwxr-xr-x 2 sergey sergey 4096 авг. 25 13:03 Документы
438064 drwxr-xr-x 2 sergey sergey 4096 авг. 25 13:03 Загрузки
438134 drwxr-xr-x 2 sergey sergey 4096 авг. 25 13:03 Изображения
438133 drwxr-xr-x 2 sergey sergey 4096 авг. 25 13:03 Музыка
438066 drwxr-xr-x 2 sergey sergey 4096 авг. 25 13:03 Общедоступные
438063 drwxr-xr-x 2 sergey sergey 4096 авг. 25 13:03 Рабочий стол
438065 drwxr-xr-x 2 sergey sergey 4096 авг. 25 13:03 Шаблоны
```

Жесткие и символьные ссылки удаляются так же, как и обычные файлы и каталоги, при этом файл, на которые они ссылаются, остаются целыми.

3. Иерархия прав доступа. Одни и те же файлы и каталоги могут быть доступны некоторому пользователю, а второму пользователю – нет. Говорят, что у второго пользователя недостаточно прав.

Говоря о правах доступа пользователя к файлам, заметим, что файлами манипулирует не сам пользователь, а запущенный им процесс (процессы будут рассматриваться в следующей лабораторной работе).

При создании объектов файловой системы каждому в обязательном порядке приписывается *ярлык*, включающий в себя UID – идентификатор пользователя-хозяина файла (каталога), GID – идентификатор группы, которой принадлежит файл (каталог), тип объекта, набор атрибутов и некоторую дополнительную информацию. *Атрибуты* определяют, кто и что имеет право делать с файлом. Посмотреть ярлыки можно командой «*ls -l*»:

```
sergey@sergey-VirtualBox:~$ ls -l
total 40
-rw-rw-r-- 2 sergey sergey 0 авг. 29 19:24 file_hard_link
lrwxrwxrwx 1 sergey sergey 8 авг. 29 19:49 file_sym_link -> file.txt
-rw-rw-r-- 2 sergey sergey 0 авг. 29 19:24 file.txt
drwxrwxr-x 2 sergey sergey 4096 авг. 28 22:00 new
drwxrwxr-x 3 sergey sergey 4096 авг. 28 22:05 newdir
drwxr-xr-x 2 sergey sergey 4096 авг. 25 13:03 Видео
drwxr-xr-x 2 sergey sergey 4096 авг. 25 13:03 Документы
drwxr-xr-x 2 sergey sergey 4096 авг. 25 13:03 Загрузки
drwxr-xr-x 2 sergey sergey 4096 авг. 25 13:03 Изображения
drwxr-xr-x 2 sergey sergey 4096 авг. 25 13:03 Музыка
drwxr-xr-x 2 sergey sergey 4096 авг. 25 13:03 Общедоступные
drwxr-xr-x 2 sergey sergey 4096 авг. 25 13:03 Рабочий стол
drwxr-xr-x 2 sergey sergey 4096 авг. 25 13:03 Шаблоны
```

Рассмотрим строчку, которая относится к файлу *file.txt*. Справа налево по порядку:

- «*file.txt*» означает имя файла;
- «авг. 29 19:24» дата и время последнего изменения файла;
- «0» размер в байтах (наш файл пуст);
- «*sergey*» группа, к которой принадлежит этот файл;
- «*sergey*» владелец этого файла;
- «2» количество жестких ссылок (в нашем случае это *file.txt* и *file_hard_link*);
- атрибуты из 10 полей.

Атрибуты представляют собой следующее. Первый слева символ говорит, что это файл («-») или папка («d»). Остальные девять полей означают следующее. Для каждого файла имеются индивидуальные права доступа, которые разбиты на три группы:

1. доступ для пользователя-владельца файла;
2. доступ для группы-владельца файла;
3. доступ для остальных пользователей.

Для каждой из этих групп устанавливаются три вида доступа: «x» - право на запуск файла, «r» - право на чтение файла, «w» - право на изменение (редактирование) файла.

Таким образом, атрибуты «-rw-rw-r--» означают, что перед нами файл, который может читать и редактировать как владелец, так и группа, а остальные пользователи могут только читать этот файл.

Авторизуемся в консоли №2 под именем *test* (если эта учетная запись была удалена, необходимо создать ее снова) и попытаемся удалить файл *file.txt*, владельцем которого является администратор (в данном случае *sergey*):

```
test@sergey-VirtualBox:/home/sergey$ cat file.txt
test@sergey-VirtualBox:/home/sergey$ rm file.txt
rm: remove write-protected regular empty file 'file.txt'? y
rm: cannot remove 'file.txt': Permission denied
```

Как видно, файл удалить не удалось в силу ограничения прав доступа

Более подробно иерархию прав доступа к файлам и папкам рассмотрим в лабораторной работе №4.

4. Команды архивирования файлов. Основным средством архивирования в Linux является комплекс из двух программ – *tar* и *gzip*. Возможно, конечно, использовать и другие программы. Программа *tar* расшифровывается как «*Tape ARchiver*», не сжимает данные, а просто объединяет разрозненные файлы в один файл с последовательным доступом. Для создания архивного файла на диске, воспользуемся командой *tar -cf <имя архивного файла> <файл 1> <файл 2> <...>*. Модификатор «c» говорит о том, что нам надо создать новый архивный файл, модификатор «f» - архив будет создан в виде файла, <файл 1> <файл 2> <...> – это те файлы, которые будут добавлены в архив.

Создадим в домашнем каталоге каталог <lab_3>, а в нем – пять текстовых файлов с именами *1.txt*, *2.txt*, *3.txt*, *4.txt*, *5.txt* произвольного содержания (размер каждого файла должен

отличаться). Добавим любые 4 из них в архив с именем *arch.tar*. Расширение *tar* можно и не указывать (но мы укажем):

```
sergey@sergey-VirtualBox:~/lab_3$ tar -cf arch.tar 1.txt 2.txt 3.txt 4.txt
sergey@sergey-VirtualBox:~/lab_3$ ls -l
total 52
-rw-rw-r-- 1 sergey sergey 1698 сент.  8 22:12 1.txt
-rw-rw-r-- 1 sergey sergey 1132 сент.  8 22:13 2.txt
-rw-rw-r-- 1 sergey sergey 2830 сент.  8 22:12 3.txt
-rw-rw-r-- 1 sergey sergey 4528 сент.  8 22:13 4.txt
-rw-rw-r-- 1 sergey sergey 8490 сент.  8 22:13 5.txt
-rw-rw-r-- 1 sergey sergey 20480 сент. 12 10:27 arch.tar
```

Для просмотра списка файлов архива нужно использовать команду *tar -tf <имя архивного файла>*:

```
sergey@sergey-VirtualBox:~/lab_3$ tar -tf arch.tar
1.txt
2.txt
3.txt
4.txt
sergey@sergey-VirtualBox:~/lab_3$
```

Для добавления отдельных файлов (не архивных) в существующий архивный файл, необходимо вместо модификатора «с» указать модификатор «r»:

```
sergey@sergey-VirtualBox:~/lab_3$ tar -rf arch.tar 5.txt
sergey@sergey-VirtualBox:~/lab_3$ tar -tf arch.tar
1.txt
2.txt
3.txt
4.txt
5.txt
sergey@sergey-VirtualBox:~/lab_3$
```

Добавить в архив можно и весь каталог. В этом случае все подкаталоги и файлы будут добавлены в архив с сохранением их структуры: *tar -cf <имя архивного файла> <имя каталога>*.

Удалить отдельные файлы из архива можно командой *tar -delete <имя удаляемого файла> -f <имя архивного файла>*:

```
sergey@sergey-VirtualBox:~/lab_3$ tar --delete 5.txt -f arch.tar
sergey@sergey-VirtualBox:~/lab_3$ tar -tf arch.tar
1.txt
2.txt
3.txt
4.txt
sergey@sergey-VirtualBox:~/lab_3$
```

После создания архивного файла его необходимо сжать. Для этого используется команда *gzip -l <имя архивного файла>*:

```
sergey@sergey-VirtualBox:~/lab_3$ gzip -l arch.tar
sergey@sergey-VirtualBox:~/lab_3$ ls -l
total 36
-rw-rw-r-- 1 sergey sergey 1698 сент.  8 22:12 1.txt
-rw-rw-r-- 1 sergey sergey 1132 сент.  8 22:13 2.txt
-rw-rw-r-- 1 sergey sergey 2830 сент.  8 22:12 3.txt
-rw-rw-r-- 1 sergey sergey 4528 сент.  8 22:13 4.txt
-rw-rw-r-- 1 sergey sergey 8490 сент.  8 22:13 5.txt
-rw-rw-r-- 1 sergey sergey  678 сент. 12 10:27 arch.tar.gz
```

Архив *arch.tar.gz* является результатом работы команды *gzip*. Модификатор «-l» указывает на степень сжатия. Он может принимать значения от 1 (минимальная степень сжатия) до 9 (максимальная степень сжатия). Сравним размеры несжатого и сжатого архивного файлов:

- не сжатый занимает 20480 байт на диске;

- сжатый занимает 678 байт на диске.

Установим максимальное сжатие (9) и посмотрим на размер архива:

```
sergey@sergey-VirtualBox:~/lab_3$ gzip -9 arch.tar
sergey@sergey-VirtualBox:~/lab_3$ ls -l
total 36
-rw-rw-r-- 1 sergey sergey 1698 сент.  8 22:12 1.txt
-rw-rw-r-- 1 sergey sergey 1132 сент.  8 22:13 2.txt
-rw-rw-r-- 1 sergey sergey 2830 сент.  8 22:12 3.txt
-rw-rw-r-- 1 sergey sergey 4528 сент.  8 22:13 4.txt
-rw-rw-r-- 1 sergey sergey 8490 сент.  8 22:13 5.txt
-rw-rw-r-- 1 sergey sergey 569 сент. 12 10:34 arch.tar.gz
```

Для просмотра информации о сжатом файле наберем команду *gzip -l <имя архивного файла>*:

```
sergey@sergey-VirtualBox:~/lab_3$ gzip -l arch.tar.gz
      compressed      uncompressed  ratio uncompressed_name
          569             20480   97.4% arch.tar
sergey@sergey-VirtualBox:~/lab_3$
```

Compressed – размер архивного файла после сжатия, *uncompressed* – размер архивного файла до сжатия, *ratio* – степень сжатия, *uncompressed_name* – имя исходного архивного файла.

Для распаковки архивного файла следует воспользоваться командой *gzip* с модификатором «-d»:

```
sergey@sergey-VirtualBox:~/lab_3$ gzip -d arch.tar.gz
sergey@sergey-VirtualBox:~/lab_3$ ls -l
total 52
-rw-rw-r-- 1 sergey sergey 1698 сент.  8 22:12 1.txt
-rw-rw-r-- 1 sergey sergey 1132 сент.  8 22:13 2.txt
-rw-rw-r-- 1 sergey sergey 2830 сент.  8 22:12 3.txt
-rw-rw-r-- 1 sergey sergey 4528 сент.  8 22:13 4.txt
-rw-rw-r-- 1 sergey sergey 8490 сент.  8 22:13 5.txt
-rw-rw-r-- 1 sergey sergey 20480 сент. 12 10:34 arch.tar
```

5. *Расщепление и восстановление* файлов. Иногда бывает полезным расщепление одного большого файла на несколько частей. Например, если исходный файл целиком не помещается на накопитель. Для этой цели служит команда *split*, которая копирует файл, разбивая его на несколько частей заданной длины. В качестве аргументов следует указать имя исходного файла и префикс имен выходных файлов. Модификатор «-b» позволяет задать размер выходных файлов в байтах. По умолчанию размер выходных файлов равен 1 Мб.

Разобьем файл с именем *File* (его спросить у преподавателя), размер которого 680 Мб, на части по 100 Мб каждая:

```
-rw----- 1 sergey sergey 681744384 июня 16 2014 File
sergey@sergey-VirtualBox:~/lab_3$ split -b100M File File_part.
```

В результате выполнения команды *File* разобьется на 7 частей одинакового размера:

```
-rw----- 1 sergey sergey 681744384 июня 16 2014 File
-rw-rw-r-- 1 sergey sergey 104857600 сент. 14 08:58 File_part.aa
-rw-rw-r-- 1 sergey sergey 104857600 сент. 14 08:59 File_part.ab
-rw-rw-r-- 1 sergey sergey 104857600 сент. 14 08:59 File_part.ac
-rw-rw-r-- 1 sergey sergey 104857600 сент. 14 08:59 File_part.ad
-rw-rw-r-- 1 sergey sergey 104857600 сент. 14 09:00 File_part.ae
-rw-rw-r-- 1 sergey sergey 104857600 сент. 14 09:00 File_part.af
-rw-rw-r-- 1 sergey sergey 52598784 сент. 14 09:00 File_part.ag
```

Для соединения частей файла следует выполнить команду *cat File_part.* >File*:

```
sergey@sergey-VirtualBox:~/lab_3$ cat File_part.* >File
```

После соединения части файла можно удалить.

ЗАДАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ

0. Подробно изучите материал, представленный в данной работе.

1. Выполните следующие операции с файлами и каталогами в консоли №1:

- 1.1. Перейдите в каталог «/», а затем обратно в домашний каталог.
- 1.2. Покажите содержимое домашнего каталога.
- 1.3. Создайте пустой текстовый файл с именем *<ваше имя>.txt* в домашнем каталоге;
- 1.4. Создайте в домашнем каталоге каталог с именем *lab_3*. Скопируйте в него текстовый файл *<ваше имя>.txt*.
- 1.5. Удалите текстовый файл *<ваше имя>.txt* из домашнего каталога.
- 1.6. Создайте не менее трех жестких ссылок на файл *<ваше имя>.txt*. Ссылки должны находиться в разных каталогах.
- 1.7. Создайте одну символическую ссылку на файл *<ваше имя>.txt*.
- 1.8. Создайте одновременно два каталога *dir1* и *dir2* в каталоге *lab_3*.
- 1.9. Скопируйте каталог *dir1* в каталог *dir2*.
- 1.10. От имени учетной записи *test* попытайтесь удалить файл *<ваше имя>.txt*.
- 1.11. Подумайте, как с помощью команды *mv* можно переименовать файл или каталог. Поделитесь вашими мыслями с преподавателем :).
- 1.12. Создайте в домашнем каталоге символическую ссылку на каталог *lab_3*.

2. Выполните операции архивирования и извлечения файлов:

- 2.1. Создайте пять текстовых файлов в каталоге *lab_3*. Размер файлов должен быть различным.
- 2.2. Создайте в каталоге *lab_3* архивный файл, состоящий из четырех текстовых файлов.
- 2.3. Добавьте в архивный файл один текстовый файл.
- 2.4. По требованию преподавателя удалите некоторые файлы из архивного файла.
- 2.5. Выполните сжатие архивного файла с различными степенями сжатия (не менее трех вариантов степеней сжатия).
- 2.6. Выполните распаковку архива.

3. Выполните операции расщепления и соединения файла:

- 3.1. Скопируйте из любого источника (можно спросить у преподавателя) файл, размером не менее 600 Мб, в каталог *lab_3*.
- 3.2. Выполните его разбиение на части по 100 Мб каждая.
- 3.3. Выполните соединение частей файла.

4. Подготовьте ответы на вопросы:

- 4.1. Дайте определение файловой системе.
 - 4.2. Перечислите основные типы файлов в файловой системе ОС Linux.
 - 4.3. Что называют домашним каталогом и какое имя он носит?
 - 4.4. Что такое индексный дескриптор?
 - 4.5. Дайте определение жесткой и символической ссылкам на файл.
 - 4.6. Можно ли задать несколько имен одной и той же папке? Подтвердите ответ примером.
 - 4.7. Что такое ярлык файловой системы? Атрибуты файла?
 - 4.8. Какие группы прав доступа к объектам файловой системы вы знаете?
5. Будьте готовы ответить на дополнительные вопросы преподавателя по данной теме.