

5407

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. В.Ф. УТКИНА

РАБОТА С ФАЙЛОВОЙ СИСТЕМОЙ LINUX

Методические указания
к лабораторным работам

Рязань 2019

УДК 004.45

Работа с файловой системой LINUX: методические указания к лабораторным работам / Рязан. гос. радиотехн. ун-т; сост.: С.А. Бубнов, А.А. Бубнов. Рязань, 2019. 28 с.

Содержат лабораторные работы по дисциплине «Операционные системы».

Предназначены для студентов очной и заочной форм обучения, изучающих курс «Операционные системы».

Ил. 16. Библиогр.: 4 назв.

Операционная система LINUX, файловая система, LVM, права доступа к объектам файловой системы

Печатается по решению редакционно-издательского совета Рязанского государственного радиотехнического университета.

Рецензент: кафедра ВПМ РГРТУ (зав. кафедрой проф. Г.В. Овечкин)

СОДЕРЖАНИЕ

| | |
|-------------------------------|----|
| ЛАБОРАТОРНАЯ РАБОТА № 1..... | 2 |
| ЛАБОРАТОРНАЯ РАБОТА № 2..... | 14 |
| ЛАБОРАТОРНАЯ РАБОТА № 3..... | 19 |
| БИБЛИОГРАФИЧЕСКИЙ СПИСОК..... | 28 |

ЛАБОРАТОРНАЯ РАБОТА № 1

ЗНАКОМСТВО С ФАЙЛОВОЙ СИСТЕМОЙ LINUX

1. Общие сведения о файловой системе. Файловая система — это структура, с помощью которой ядро операционной системы предоставляет пользователям (процессам) ресурсы долговременной памяти системы. *Файл* в системе LINUX — это любая последовательность байт, содержащая произвольную информацию. Нет различия между текстовыми, двоичными и любыми другими типами файлов. Расширение файла носит весьма условный характер. Файл может иметь несколько расширений. Для удобства файлы группируются в *каталоги* — это файлы специального типа. Каталоги могут содержать подкаталоги, что приводит к иерархической файловой системе. В LINUX корневой каталог обозначается «/». Фрагмент корневой файловой системы приведен на рис. 1 (самый правый столбец содержит имена каталогов и файлов, находящихся в корневом каталоге).

```

drwxr-xr-x  2 root root 12288 авг 12 13:29 bin
drwxr-xr-x  3 root root  4096 авг 12 14:00 boot
drwxrwxr-x  2 root root  4096 окт 13  2016 cdrom
drwxr-xr-x 19 root root  3980 сен 22 13:07 dev
drwxr-xr-x 134 root root 12288 сен 22 12:46 etc
drwxr-xr-x  6 root root  4096 сен 22 09:53 home
lrwxrwxrwx  1 root root    32 авг 12 13:47 initrd.img -> boot/initrd.img-4.4.0-$
lrwxrwxrwx  1 root root    33 авг 12 12:29 initrd.img.old -> boot/initrd.img-3.$
drwxr-xr-x 24 root root  4096 авг 12 13:59 lib
drwx----- 3 root root 16384 окт 13  2016 lost+found
drwxr-xr-x  4 root root  4096 сен 20 10:50 media
drwxr-xr-x  2 root root  4096 авг 12 13:16 mnt
drwxr-xr-x  3 root root  4096 авг 13 21:52 opt
dr-xr-xr-x 199 root root    0 сен 20 11:24 proc
drwx----- 5 root root  4096 сен 11 18:02 root
drwxr-xr-x 28 root root   880 сен 22 12:53 run
drwxr-xr-x  2 root root 12288 авг 13 21:53/sbin
drwxr-xr-x  2 root root  4096 апр 29 11:39 snap

```

Рис. 1. Содержимое корневой файловой системы Linux

ОС LINUX может работать с различными типами файловых систем: *Ext2*, *Ext3*, *Ext4*, *JFS*, *XFS*, *Btrfs*, *ZFS*, а также с *FAT* и *NTFS* системами.

Каждому файлу соответствует *индексный дескриптор (inode)*. Он содержит всю необходимую файловой системе информацию о файле:

- тип файла (обычный файл, каталог, файл-дырка и т.д.);
- права доступа к файлу;
- информацию о владельце файла;

- 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 последовательно и содержатся в специальной таблице, которая формируется при создании файловой системы на носителе. Каждый логический или физический диск имеет собственную таблицу индексных дескрипторов. Номер индексного дескриптора файла является его непосредственным именем в системе. Таким образом, для операционной системы имя любого файла в LINUX — это ссылка на номер его индексного дескриптора.

Пользователю неудобно использовать индексный дескриптор в качестве имени файла. Поэтому в ОС предусмотрено создание имен, понятных пользователю, которые связаны с индексными дескрипторами посредством *жестких ссылок*.

Жесткая ссылка — это связь типа «имя файла — номер индексного дескриптора». Для любого файла можно задать несколько жестких ссылок, т.е. имен. Индексный дескриптор файла хранит информацию о количестве жестких ссылок. Жесткая ссылка связана непосредственно с номером индексного дескриптора, поэтому она может указывать только на файлы, находящиеся внутри той же файловой системы. Нельзя создать жесткую ссылку на файл, расположив ее на другом логическом или физическом дисках по отношению к этому файлу. Чтобы полностью удалить файл из системы, необходимо удалить **все** жесткие ссылки на него.

Также используется другой тип ссылки — *символическая ссылка* — это файл, содержащий путь к другому файлу или каталогу. Если удалить файл, на который указывает символическая ссылка, то она станет бесполезной. Символические ссылки могут указывать на файлы и каталоги, расположенные в других файловых системах (на других логических или физических дисках).

Имя файла состоит из пути к файлу начиная от корневого каталога и собственного имени файла (абсолютный путь). Например, `/home/main/file_1.txt` - `/home/main` — путь к файлу, `file_1.txt` — его собственное имя.

Рабочий каталог — это каталог, в котором в данный момент

работает пользователь. Указав путь к файлу относительно рабочего каталога, получим относительный путь.

Точка, являющаяся первым символом в имени файла, свидетельствует о том, что данный файл является *скрытым* (например, «.file_1»). В каждом каталоге обязательно присутствуют два специальных файла с именами «.» и «. .» (точка и две точки). Файл с именем «.» является ссылкой на текущий каталог, файл с именем «. .» - ссылкой на родительский каталог.

2. Работа с файлами и каталогами. Определить рабочий каталог можно командой *pwd*. На экран выводится абсолютное имя рабочего каталога.

Для смены каталога используется команда *cd [имя_каталога]*. В качестве имени каталога можно использовать как его абсолютное имя, так и относительное. Например, из текущего каталога */home/main* можно попасть в каталог */home/main/Общедоступные* двумя способами, показанными на рис. 2.

```
main@main-VirtualBox:~$ pwd
/home/main
main@main-VirtualBox:~$ cd /home/main/Общедоступные
main@main-VirtualBox:~/Общедоступные$ cd ..
main@main-VirtualBox:~$ cd Общедоступные
main@main-VirtualBox:~/Общедоступные$
```

Рис. 2. Переход между каталогами

Создать пустой файл в текущем каталоге (если указано относительное имя файла) можно командой *touch <имя_файла>*.

Команда *cat > <имя_файла>* позволяет создать файл и наполнить его текстом, набранным с клавиатуры (использована команда перенаправления вывода «>») (рис. 3).

```
main@main-VirtualBox:~$ cat > file
Hello!
main@main-VirtualBox:~$ cat file
Hello!
main@main-VirtualBox:~$
```

Рис. 3. Наполнение текстового файла содержимым

Для просмотра содержимого текстовых файлов можно воспользоваться командой *cat [-опции] [вх_файл_1[вх_файл_2...]]*. Предназначение команды *cat* — конкатенация (объединение) нескольких файлов и вывод результата на стандартный вывод (экран). Для просмотра многостраничных файлов существует команда *more <имя_файла>*.

Для просмотра содержимого каталога служит утилита *ls [-опции] [<имя_каталога>]*. Наиболее часто используемые опции

утилиты *ls* приведены в таблице.

| Команда [-опции] | Действие |
|------------------------------|---|
| <i>ls</i> <имя_каталога> | Просмотр содержимого каталога <i>имя_каталога</i> |
| <i>ls -a</i> | Вывод на экран списка всех файлов и каталогов рабочего каталога, включая скрытые и специальные |
| <i>ls -l</i> | Вывод на экран списка всех файлов и каталогов рабочего каталога в виде таблицы. Специальные файлы и скрытые файлы и каталоги не отображаются. Также отображаются права доступа и тип файлов (файл, каталог, канал и т.д.) |
| <i>ls -la</i> | Объединение результатов команд <i>ls -a</i> и <i>ls -l</i> |
| <i>ls -li</i> | Вывод на экран списка всех файлов и каталогов рабочего каталога, включая номера индексных дескрипторов |
| <i>ls -l > ls_out.txt</i> | Результат команды <i>ls -l</i> сохраняется в файл с именем <i>ls_out.txt</i> в рабочем каталоге |

Для определения типа файла предназначена утилита *file* [-опции] <имя_файла>. На рис. 4 показаны результаты работы утилиты с файлами различных типов.

```
main@main-VirtualBox:~/d_1$ file file_{dir,1,2,3,4,5}
file_dir: directory
file_1:  ASCII text
file_2:  ASCII text
file_3:  symbolic link to /home/main/d_1/file_1
file_4:  Audio file with ID3 version 2.4.0, contains: MPEG ADTS, layer III, v1,
320 kbps, 44.1 kHz, Stereo
file_5:  PDF document, version 1.4
main@main-VirtualBox:~/d_1$
```

Рис. 4. Результат действия команды *file*

Используя опцию *-s*, можно получить информацию об устройствах, с которыми работает данная операционная система (поскольку в LINUX все устройства отображаются в виде файлов). Например, получить информацию о трех разделах жесткого диска, которому соответствует файл */dev/sda*, можно командой *file -s /dev/sda{1,2,3}*.

Для сравнения содержимого двух файлов побайтно используется утилита *cmp* [-опции] <файл_1> <файл_2>. Если

файл_1 и *файл_2* полностью идентичны, то утилита просто заканчивает свою работу без каких-либо сообщений. В таблице показаны результаты работы утилиты с некоторыми опциями.

| Команда [-опции] | Действие |
|---|---|
| <i>стр</i> < <i>файл_1</i> > < <i>файл_2</i> > | Вывод на экран номера строки и номера байта в строке, где имеется первое различие |
| <i>стр -b</i> < <i>файл_1</i> > < <i>файл_2</i> > | Вывод на экран номера строки и номера байта в строке, где имеется первое различие, а также ASCII-коды первых различающихся символов и сами символы |
| <i>стр -l</i> < <i>файл_1</i> > < <i>файл_2</i> > | Вывод на экран списка всех различающихся байтов, а также ASCII-коды всех различающихся символов |

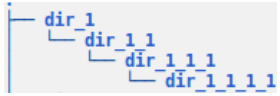
Более подробную информацию о сравниваемых файлах выводит на экран утилита *diff* [-опции] <*файл_1*> <*файл_2*>.

Для создания символических и жестких ссылок служит утилита *ln* [-опции] <*файл_цель*> <*имя_ссылки*>. Варианты использования данной команды приведены в таблице.

| Команда [-опции] | Действие |
|---|--|
| <i>ln -s</i> < <i>файл_цель</i> > < <i>имя_ссылки</i> > | Создание символической ссылки |
| <i>ln</i> < <i>файл_цель</i> > < <i>имя_ссылки</i> > | Создание жесткой ссылки |
| <i>ln -P</i> < <i>симв_ссылка</i> > < <i>имя_ссылки</i> > | Создание жесткой ссылки на символьную ссылку |

Далее в таблице рассмотрены наиболее употребительные команды работы с файлами и каталогами.

| Команда [-опции] | Действие |
|--|--|
| <i>Создание и удаление</i> | |
| <i>mkdir</i> < <i>имя_каталога_1</i> [<i>имя_каталога_2...</i>]> | Создание одного или нескольких каталогов |
| <i>mkdir -p</i> <i>dir_1/dir_1_1/dir_1_1_1/dir_1_1_1_1</i> | Создание дерева каталогов |

| | |
|---|--|
| |  |
| <code>rm <имя_файла></code> | Удаление файла |
| <code>rm -r <имя_каталога></code> | Рекурсивное удаление каталога и его содержимого |
| Копирование и перемещение | |
| <code>cp <вх_файл> <вых_файл></code> | Копирование файла <code>вх_файл</code> . Имя скопированного файла - <code>вых_файл</code> |
| <code>cp <вх_файл_1[вх_файл_2...]> <имя_каталога></code> | Копирование файлов <code>вх_файл_1</code> , <code>вх_файл_2</code> ... в каталог <code>имя_каталога</code> |
| <code>cp -r <вх_каталог> <вых_каталог></code> | Рекурсивное копирование |
| <code>mv <вх_файл (каталог)> <вых_каталог></code> | Перемещение файла или каталога в указанный каталог |
| <code>mv -i <вх_файл> <вых_файл></code> | Перемещение файла <code>вх_файл</code> в <code>вых_файл</code> с подтверждением на перезапись |
| <code>mv -n <вх_файл_1> <вых_файл></code> | Перемещение без перезаписи существующих файлов |

3. Архивирование и сжатие файлов. Для объединения нескольких файлов и каталогов в один архивный файл используется утилита `tar` [`-опции`] `<имя_архивного_файла>` `<файл_1>` `<файл_2>`...`<файл_n>`. По умолчанию архивный файл создается на ленточном устройстве `/dev/rmt0`. Для создания архивного файла на диске следует использовать опцию `-f`, после которой следует указать имя архивного файла. Некоторые варианты использования данной утилиты приведены в таблице.

| Команда [-опции] | Действие |
|---|---|
| <i>tar -cf</i> <имя_архива> <файл_1 [файл_2...]> | Создание архивного файла с именем <имя_архива>, который содержит в себе файлы, указанные в качестве аргументов |
| <i>tar -Af</i> <имя_архива> <архив_1 [архив_2...]> | Добавление новых архивных файлов в существующий архивный файл |
| <i>tar -tf</i> <имя_архива> <файл_1 [файл_2...]> | Вывод на экран (по умолчанию) списка всех файлов, содержащихся в архиве |
| <i>tar -- delete -f</i> <имя_архива> <файл_1 [файл_2...]> | Удаление из архива указанных файлов |
| <i>tar -rf</i> <имя_архива> <файл_1 [файл_2...]> | Добавление в конец архивного файла новых файлов |
| <i>tar -xf</i> <имя_архива> | Извлечение файлов из архива в рабочий каталог |
| <i>tar -cvf</i> <имя_архива> <./каталог*> | Добавление в архивный файл всех файлов из каталога <i>каталог</i> , а также всех файлов из подкаталогов каталога <i>каталог</i> |

Поскольку рассмотренная выше утилита только архивирует файлы, не сжимая их, то для сжатия архивного файла применяются утилиты *gzip* [-опции] <имя_архивного_файла> и *bzip2* [-опции] <имя_архивного_файла>. Утилита *gzip* для сжатия данных использует алгоритм Лемпела – Зива. Утилита *bzip2* для сжатия использует алгоритм Барроуза – Уилера, а затем алгоритм Хаффмана.

Примеры использования утилиты *gzip* приведены в таблице.

| Команда [-опции] | Действие |
|----------------------------------|---|
| <i>gzip [-1..9]</i> <имя_архива> | Сжатие архивного файла со степенью сжатия от 1 до 9. При этом входной архивный файл удаляется |
| <i>gzip -l</i> <имя_архива> | Вывод на экран (по умолчанию) информации о сжатом архиве |
| <i>gzip -t</i> <имя_архива> | Проверка на целостность архивного |

| | файла |
|-----------------------------|---|
| <i>gzip -d</i> <имя_архива> | Распаковка архивного файла в рабочий каталог. При этом сжатый архивный файл удаляется |
| <i>gzip -k</i> <имя_архива> | Запрет на удаление исходных файлов после сжатия или распаковки |

Примеры использования утилиты *bzip2* приведены в таблице.

| Команда [-опции] | Действие |
|------------------------------------|--|
| <i>bzip2</i> [-1..9] <имя_архива> | Сжатие архивного файла со степенью сжатия от 1 до 9. При этом исходный архивный файл удаляется |
| <i>bzip2 -t</i> <имя_архива> | Проверка на целостность архивного файла |
| <i>bzip2 -k</i> <имя_архива> | Запрет на удаление исходных файлов после сжатия или распаковки |
| <i>bzip2 -z</i> <имя_архива> | Принудительное сжатие архивного файла |
| <i>bzip2 -d</i> <имя_архива> | Принудительная распаковка |

Поясним необходимость применения двух последних опций из таблицы. По умолчанию утилита *bzip2* автоматически определяет действие над файлом в зависимости от его суффикса: сжатие, если имя файла заканчивается не на *.bz2*, распаковка, если имя файла заканчивается на *.bz2*. Опции *-z* и *-d* отключают автоматическое определение.

4. Расщепление и восстановление файлов. Для разбиения файла на части служит утилита *split* [-опции] <файл_для_расщепления>. Данная утилита копирует файл, разбивая его на части. Для облегчения поиска полученных частей можно задать префикс. В таблице показаны некоторые варианты использования утилиты *split*.

| Команда [-опции] | Действие |
|--------------------------------|--|
| <i>split -b10b</i> <имя_файла> | Расщепление файла на части, размер каждой равен 10 байт |
| <i>split -b10K</i> <имя_файла> | Расщепление файла на части, размер каждой равен 10*1024 байт |
| <i>split -b10M</i> <имя_файла> | Расщепление файла на части, размер |

| | |
|------------------------------------|---|
| <i><префикс></i> | каждой равен 10*1024 Кбайт. Имя каждой части начинается на <i>префикс</i> |
| <i>split -n5 <имя_файла></i> | Расщепление файла на 5 частей |

Для конкатенации (объединения) получившихся частей можно использовать утилиту *cat [-опции] <вх_файл_1[вх_файл_2...]>*. На рис. 5 показан пример расщепления файла на блоки. Размер каждого блока — 1 Мбайт, размер исходного файла - 8079326 байт, префикс — *file_4_*.

```
main@main-VirtualBox:~/lab_4/type$ split -b1M file_4 file_4_
main@main-VirtualBox:~/lab_4/type$ ls -l|grep file_4_
-rw-rw-r-- 1 main main 1048576 окт 9 15:17 file_4_aa
-rw-rw-r-- 1 main main 1048576 окт 9 15:17 file_4_ab
-rw-rw-r-- 1 main main 1048576 окт 9 15:17 file_4_ac
-rw-rw-r-- 1 main main 1048576 окт 9 15:17 file_4_ad
-rw-rw-r-- 1 main main 1048576 окт 9 15:17 file_4_ae
-rw-rw-r-- 1 main main 1048576 окт 9 15:17 file_4_af
-rw-rw-r-- 1 main main 1048576 окт 9 15:17 file_4_ag
-rw-rw-r-- 1 main main 739294 окт 9 15:17 file_4_ah
```

Рис. 5. Результат работы утилиты *split*

Процедура конкатенации имеет вид *cat file_4_* > file_4_conc*.

Чтобы не перечислять все объединяемые файлы, можно воспользоваться шаблоном: *file_4_** - все файлы, имеющие префикс *file_4_*. Результат работы утилиты *cat* перенаправим в файл с именем *file_4_conc*.

5. Монтирование файловой системы. Чтобы операционная система могла работать с подключенными носителями (жесткий диск, образ диска, флэш-накопитель, сетевая папка и т. д.), необходимо провести операцию *монтирования файловой системы*. В *UNIX*-подобных системах нет понятий «логический диск», «буква диска» и т. д. Вместо этого к корневой файловой системе следует подключить (смонтировать) новую файловую систему. При этом не важно, на каком носителе она будет расположена физически. После проведения операции монтирования в системе появится каталог, отражающий содержимое подключенного носителя.

Рассмотрим процедуры монтирования и демонтирования файловой системы на примере подключения нового жесткого диска *ADD_DISC* к SATA-контроллеру жестких дисков.

Файлы, посредством которых осуществляется взаимодействие операционной системы с устройствами, находятся в каталоге */dev*. Буквенное обозначение жестких дисков зависит от контроллера, к которому они подключены. Например,

- *sd* — устройство SCSI;

- *hd* — устройство ATA;
- *vd* — виртуальное устройство;
- *mmcblk* — флэш-накопители, подключенные через карт-ридер.

Выполнив команду `ls -l /dev | grep sd`, можно узнать буквенно-цифровое обозначение подключаемого жесткого диска (рис. 6).

```
main@main-VirtualBox:~/Рабочий стол$ ls -l /dev | grep sd
brw-rw---- 1 root disk 8, 0 окт 9 16:01 sda
brw-rw---- 1 root disk 8, 1 окт 9 16:01 sda1
brw-rw---- 1 root disk 8, 2 окт 9 16:01 sda2
brw-rw---- 1 root disk 8, 3 окт 9 16:01 sda3
brw-rw---- 1 root disk 8, 16 окт 9 16:01 sdb
```

Рис. 6. Буквенно-цифровое обозначение дисков SCSI

Видно, что диск *sda* разбит на три раздела. Это основной жесткий диск. В последней строке находится устройство *sdb* — это подключенный к контроллеру SCSI жесткий диск.

Создать файловую систему на подключаемом диске можно с помощью команды `mkfs -t ext4 /dev/sdb`. Тип создаваемой файловой системы — *ext4*. Утилиту необходимо запустить с правами суперпользователя (рис. 7).

```
main@main-VirtualBox:~/Рабочий стол$ sudo mkfs -t ext4 /dev/sdb
[sudo] password for main:
mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: c9ecbec7-b44a-4948-9be8-42fb82b9af28
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

Рис. 7. Создание файловой системы на подключаемом диске

Далее необходимо создать точку монтирования, т. е. каталог, который будет отображать содержимое подключаемого диска. Расположим точку монтирования в домашнем каталоге и дадим имя *mnt*: `mkdir ~/mnt`.

Последний шаг — монтирование файловой системы. Для этого следует выполнить команду `sudo mount <устройство> <точка_монтирования>`. Для отключения диска служит утилита `umount <устройство>`. На рис. 8 показаны подключение и отключение файловой системы.

```

main@main-VirtualBox:~$ sudo mount /dev/sdb ~/mnt
main@main-VirtualBox:~$ ls -l ~/mnt
total 16
drwx----- 2 root root 16384 окт 10 22:43 lost+found
main@main-VirtualBox:~$ sudo umount /dev/sdb
main@main-VirtualBox:~$

```

Рис. 8. Подключение и отключение файловой системы (диска)

Заметим, что после монтирования система меняет права доступа к каталогу `~/mnt`. Владелец (суперпользователь) имеет полный доступ, а группа и все остальные — только чтение и просмотр каталога.

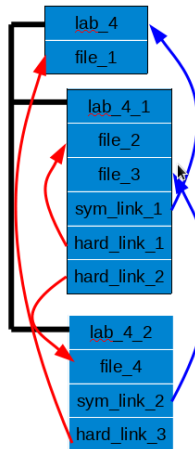
Примеры использования утилиты `mount` приведены в таблице.

| Команда [-опции] | Действие |
|---|---|
| <code>mount <файл-устройство> <точка_монтирования></code> | Монтирование устройства в указанную точку монтирования |
| <code>mount -l</code> | Отображение списка всех примонтированных файловых систем |
| <code>mount -B <olddir> <newdir></code> | Монтирование каталога <code>olddir</code> в каталог <code>newdir</code> |
| <code>mount -U <uuid></code> | Монтирование раздела диска по его UUID |
| <code>umount <файл-устройство></code> | Демонтирование файловой системы |

ЗАДАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ

1. Изучить теоретический материал.
2. Авторизоваться в консоли № 1.
3. Определить рабочий (текущий) каталог.
4. Перейти в корневой каталог и вернуться обратно.
5. Вывести содержимое домашнего каталога на экран в виде таблицы.
6. Повторить предыдущее задание, добавив к выходному листингу скрытые файлы и каталоги. Пояснить значение каждого поля на примере конкретного файла или каталога.
7. В домашнем каталоге создать каталог `lab_4` и поместить в него архивный файл `files_lab_4.tar`, приложенный к описанию лабораторной работы.

8. В каталоге *lab_4* создать текстовые файлы с именами *file_1* *file_2* и заполнить их произвольными символами в несколько строк (3-5 строк по 15-20 символов в каждой). При этом в обоих файлах должно быть хотя бы по одной одинаковой строке. Выполнить сравнение содержимого этих файлов утилитами *cmp* и *diff*. Выводимые результаты объяснить.



hard_link {1,2,3} — жесткие ссылки на объекты, к которым ведут красные стрелки

sym_link {1,2} — символические ссылки на объекты, к которым ведут синие стрелки

lab {4,4 1,4 2} — каталоги

file {1,2,3,4} — файлы (могут быть пустыми)

9. В каталоге *lab_4* создать каталоги *lab_4_1* и *lab_4_2*. В каталоге *lab_4_1* создать пустые файлы *file_2* и *file_3*. В каталоге *lab_4_2* создать пустой файл *file_4*. Далее создать символичные и жесткие ссылки, как показано на рисунке.
10. Изучить работу команд копирования и перемещения файлов и каталогов.
11. Распаковать архивный файл *files_lab_4.tar* с помощью утилиты *tar*.
12. Воспользоваться утилитой *file* для определения типа каждого файла, находящегося в каталоге *files_lab_4*. Каталог *files_lab_4* образуется после распаковки архивного файла *files_lab_4.tar*.
13. Создать с помощью утилиты *tar* в каталоге *lab_4* архивный файл с именем *arch_file*, содержащий:
- все файлы из каталога *files_lab_4*;
 - архивный файл *files_lab_4.tar*;
 - файлы *sym_link_1*, *sym_innk_2*, *hard_link_1*.
14. Исследовать работу утилит *gzip*, *bzip2* на созданном в п. 13 архивном файле. Сравнить размеры сжатых архивных файлов. Сделать вывод о максимальной степени сжатия каждой утилиты.
15. Проверить на целостность полученные архивные файлы.
16. Из каталога *files_lab_4* выбрать самый большой по размеру файл и расщепить его:
- на 150 частей с префиксом *part_1_*;
 - на блоки размером 200 Кбайт с префиксом *part_2*;
 - выполнить конкатенацию частей с префиксом *part_1* в файл с

- именем *result_1*, а с префиксом *part_2* — с именем *result_2*.
17. Монтирование жесткого диска:
 - создать виртуальный жесткий диск фиксированного типа размером 1 Гбайт. Подключить его к SATA контроллеру;
 - создать на нем файловую систему *ext3*;
 - создать точку монтирования *~/mnt_HDD*;
 - осуществить операции монтирования и демонтирования жесткого диска.
 18. Утилитой *mount* подключить файл образа *.iso* к файловой системе в точке монтирования *~/mnt_iso* и просмотреть его содержимое. Файл образа взять у преподавателя.
 19. Утилитой *mount* смонтировать *usb-накопитель* в точке монтирования *~/mnt_usb*, подключенный к одному из *usb-портов* компьютера. Для подключения *usb-накопителя* к виртуальной машине следует выполнить команду *Устройства/USB/<накопитель>*.
 20. Утилитой *mount* смонтировать каталог *~* в каталог *~/mnt*. Результаты объяснить преподавателю.
 21. В пустом каталоге *~/mnt* создать несколько файлов (команды *touch* либо *cat*). Смонтировать в этот каталог *usb-накопитель*. Просмотреть содержимое каталога *~/mnt*. Демонтировать *usb-накопитель*. Снова просмотреть содержимое каталога *~/mnt*. Результаты пояснить.

ЛАБОРАТОРНАЯ РАБОТА № 2

ПРАВА ДОСТУПА К ФАЙЛАМ И КАТАЛОГАМ В LINUX

Доступ к объектам файловой системы. Поскольку LINUX-системы являются многопользовательскими, то ограничение прав доступа одних пользователей к файлам и каталогам других пользователей имеет весьма важное значение. Каждый зарегистрированный в системе пользователь снабжается идентификатором *UID*, а зарегистрированная группа — идентификатором *GID*. **Каждый файл или каталог имеет двух владельцев: владельца-пользователя и владельца-группу.** Этого достаточно для организации разграничения прав доступа.

Факт использования файла процессом называется *доступом к файлу*, а способ воспользоваться файлом — *видом доступа*. При создании файла или каталога владельцем становится процесс, его создавший. А поскольку все процессы снабжены идентификатором *UID*, то владельцем является тот пользователь, от имени которого был запущен этот процесс. Владелец-группа (по умолчанию) — первичная

группа владельца-пользователя.

Доступ к объектам файловой системы регулируется с помощью **эффективных** идентификаторов *EUID* и *EGID*. Например, использование утилиты *sudo* позволяет изменять реальные идентификаторы на эффективные. Поэтому обычные пользователи могут выполнять многие команды от имени суперпользователя.

Создадим пустые файлы *file_1* и *file_2*:

- *file_1* с помощью команды *touch file_1*;

- *file_2* с помощью команды *sudo touch file_2*. Результаты показаны на рис. 1.

```
main@main-VirtualBox:~/lab_5$ touch file_1; sudo touch file_2; ls -l
total 56
-rw-rw-r-- 1 main main      0 окт 28 12:49 file_1
-rw-r--r-- 1 root root      0 окт 28 12:49 file_2
```

Рис. 1. Созданные файлы имеют разных владельцев

Владелец-пользователь файла *file_1* — пользователь *main*, владелец-группа — *main*. Владелец-пользователь файла *file_2* — *root*, владелец-группа — *root*. Процесс создания файла *file_2* был запущен от имени суперпользователя, поэтому и сам файл будет ему принадлежать. В данном примере идентификаторы пользователя *main* (*UID=1000* и *GID=1000*) на некоторое время были изменены на идентификаторы пользователя *root* (*UID=0* и *GID=0*).

Каждый файл или каталог файловой системы имеют атрибуты, среди которых имеются индивидуальные права доступа, которые разбиты на три категории:

- доступ для пользователя-владельца (первая триада бит *rw*x);
- доступ для группы-владельца (вторая триада бит *rw*x);
- доступ для остальных пользователей (третья триада бит

*rw*x).

Каждой категории назначаются три вида доступа:

- *r* — право на чтение (*read*);
- *w* — право на запись (*write*);
- *x* — право на выполнение (*execute*).

Если какой-либо вид доступа разрешен, то в соответствующем поле устанавливается символ «*r*», «*w*» или «*x*». Если какой-то вид доступа запрещен, то в соответствующем поле устанавливается символ «-».

Просмотреть права доступа к файлам или каталогам позволяет утилита *ls -l* (рис. 2).

```

main@main-VirtualBox:~/lab 5$ ls -l
total 4
-rw-rw-r-- 1 main main 0 окт 28 12:49 file_1
-rw-r--r-- 1 root root 0 окт 28 13:13 file_1
-rw-r--r-- 1 root root 0 окт 28 12:49 file_2
-rwxrwxrwx 1 main main 82 окт 27 11:49 process

```

Рис. 2. Отображение прав доступа

Права доступа могут иметь *числовое* и *символическое* выражения.

При назначении прав *числовым* выражением каждая категория прав доступа обозначается трехразрядным двоичным числом. При этом «1» соответствует наличию права, «0» - отсутствию такового. Получившиеся для каждой категории двоичные числа, необходимо перевести в восьмеричные и составить из них трехразрядное восьмеричное число. На рис. 3 показано получение числового выражения прав (восьмеричного числа) некоторого объекта файловой системы.

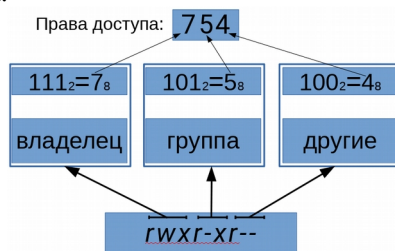


Рис. 3. Формирование прав доступа

Вид доступа к файлам отличается от вида доступа к каталогам. В таблице показаны эти отличия.

| Вид доступа | Файл | Каталог |
|-------------|--|---|
| Чтение | Разрешаются чтение содержимого файла, копирование файла | Разрешается просмотр списка содержимого каталога. Чтение, запуск, редактирование файлов из данного каталога запрещены |
| Запись | Разрешается редактировать содержимое файла. Сам файл удалять и | Разрешается создавать в данном каталоге новые файлы и каталоги, копировать в него файлы и |

| | | |
|------------|---|--|
| | переименовывать нельзя (определяется правами доступа на каталог, в котором этот файл находится) | каталоги из других каталогов, переименовывать и удалять файлы |
| Выполнение | Разрешается запускать файл, если он является файлом сценария (скриптом) | Разрешает сделать каталог текущим и выполнять с файлами и каталогами, находящимися внутри него, любые действия, разрешенные правами для самих файлов и каталогов |

Для изменения владельца-пользователя объекта применяется утилита *chown* [-опции] <имя_файла_(каталога)>. Варианты использования этой утилиты приведены в таблице.

| Команда [-опции] | Действие |
|--|---|
| <i>chown</i> <имя_нового_владельца> <имя_файла> | Устанавливает файлу с именем <имя_файла> пользователя-владельца с именем <имя_нового_владельца> |
| <i>chown</i> <имя_нового_владельца> <имя_каталога> | Устанавливает каталогу с именем <имя_каталога> пользователя-владельца с именем <имя_нового_владельца> |
| <i>chown</i> -R <имя_нового_владельца> <имя_каталога> | Устанавливает каталогу с именем <имя_каталога> и всему его содержимому пользователя-владельца с именем <имя_нового_владельца> |
| <i>chown</i> <имя_нового_владельца>: <имя_новой_группы> <имя_файла_(каталога)> | Устанавливает файлу (каталогу) с именем <имя_файла> нового пользователя-владельца и новую группу-владельца |

Для смены владельца-группы объекта файловой системы применяется утилита *chgrp* [-опции] <имя_файла_(каталога)>. Варианты использования — в таблице.

| Команда [-опции] | Действие |
|--|--|
| <i>chgrp</i> <имя_новой_группы> <имя_файла> | Устанавливает файлу с именем <имя_файла> пользователя-группу с именем <имя_новой_группы> |
| <i>chgrp</i> <имя_новой_группы> <имя_каталога> | Устанавливает каталогу с именем <имя_каталога> пользователя-группу с именем <имя_новой_группы> |
| <i>chgrp -R</i> <имя_новой_группы> <имя_каталога> | Устанавливает каталогу с именем <имя_каталога> и всему его содержимому пользователя-группу с именем <имя_новой_группы> |

Для установки прав доступа на объекты файловой системы применяется утилита *chmod* [-опции] <числовое_выражение_прав> <имя_файла_(каталога)>. Варианты использования — в таблице.

| Команда [-опции] | Действие |
|------------------------------------|--|
| <i>chmod</i> 744 <имя_файла> | Устанавливает файлу с именем <имя_файла> права доступа 744 |
| <i>chmod</i> 744 <имя_каталога> | Устанавливает каталогу с именем <имя_каталога> права доступа 774 |
| <i>chmod -R</i> 744 <имя_каталога> | Устанавливает каталогу с именем <имя_каталога> и всему его содержимому права доступа 774 |

Кроме бит доступа каждый объект файловой системы LINUX имеет еще дополнительные биты. Среди них — так называемый «Sticky-bit» («липкий бит»). Обычно этот бит используется для каталогов и означает следующее. Когда «Sticky-bit» установлен для каталога, то удалять и переименовывать файлы и подкаталоги, находящиеся в данном каталоге, могут только их владельцы или суперпользователь. Установить «Sticky-bit» можно командой *chmod 1*<числовое_выражение_прав> <имя_каталога>. При этом в третьей триаде бит вместо символа «x» появится символ «t». Например, для каталога установим «Sticky-bit», остальные права доступа оставив без изменения (рис. 4).

```

До установки Sticky-bit

main@main-VirtualBox:~/main$ ls -l | grep lab_5
drwxr-xr-x 3 main main 4096 окт 31 12:16 lab_5

После установки Sticky-bit

main@main-VirtualBox:~/main$ chmod 1755 lab_5
main@main-VirtualBox:~/main$ ls -l | grep lab_5
drwxr-xr-t 3 main main 4096 окт 31 12:16 lab_5

```

Рис. 4. Установка *Sticky-bit*

ЗАДАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ

1. Изучить теоретический материал.
2. Авторизоваться в консоли № 1.
3. В каталоге `~/lab_4` создать объекты файловой системы согласно таблице. **Примечание:** *user_1* — пользователь, созданный при установке ОС; *user_2* — пользователь, созданный командой *adduser*; *group_1* - первичная группа пользователя *user_1*; *group_2* — первичная группа пользователя *user_2*. При выполнении задания удобнее в первой консоли авторизоваться как *user_1*, а во второй консоли — как *user_2* и клавишами **CTRL+ALT+F1(F2)** переключаться между ними.

| Имя | Тип | Владелец-пользователь | Владелец-группа | Права доступа | Sticky-bit |
|---------------|---------|-----------------------|-----------------|------------------|------------|
| <i>file_1</i> | файл | <i>user_1</i> | <i>group_1</i> | <i>rw--wx-w-</i> | нет |
| <i>file_2</i> | файл | <i>user_2</i> | <i>group_2</i> | <i>r-xr-xr--</i> | нет |
| <i>common</i> | каталог | <i>user_1</i> | <i>group_2</i> | <i>rw-rwxr-x</i> | есть |

4. Продемонстрировать работу *Sticky-bit* для каталога *common*.
5. Освоить работу с утилитами *chown*, *chgrp*, *chmod*.
6. На примере созданных в п. 3 объектов файловой системы продемонстрировать умения различать виды доступа для файлов и каталогов.

ЛАБОРАТОРНАЯ РАБОТА № 3

ЗНАКОМСТВО С LVM В LINUX

1. Общие сведения о LVM. Менеджер логических томов (*LVM - Logical Volume Manager*) позволяет управлять дисковым пространством, абстрагируясь от физических устройств (дисков). У логических томов, созданных с помощью *LVM*, можно легко

изменить размер, а их названия могут нести большую смысловую нагрузку, в отличие от традиционных `/dev/sda`, `/dev/hd` либо `UUID`.

LVM позволяет собрать в один «диск» множество разнородных дисков, а затем разбить его на необходимое количество разделов.

Модель представления дискового пространства *LVM* показана на рис. 1.

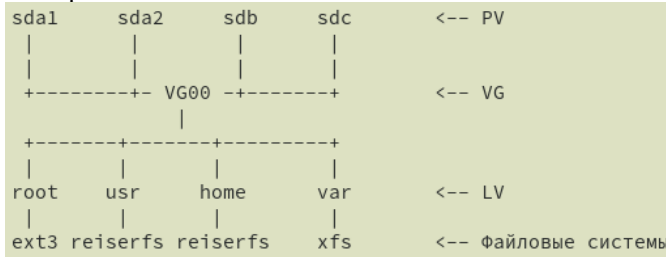


Рис. 1. Уровни LVM

Физический том (*PV — Physical Volume*) состоит из физических дисков (*sdb*, *sdc*) или разделов дисков (*sda1*, *sda2*). Каждый физический том делится на порции данных - *физический экстен*т (*PE — Physical Extent*). Каждый физический экстент имеет уникальный номер в пределах физического тома.

Группа томов (*VG — Volume Group*) — включает в себя физические тома и является верхним уровнем модели *LVM*. Представляет собой единую административную единицу.

Логический том (*LV — Logical Volume*) — представляет собой раздел группы томов. Представляет собой блочное устройство и, следовательно, может содержать файловую систему. Каждый логический том делится на порции данных - *логический экстен*т (*LE — Logical Extent*).

Размер логического экстента равен размеру физического экстента. Между физическими и логическими экстентами существует взаимно-однозначное соответствие, т. е. *LE* отображаются на *PE*. Существуют два алгоритма отображения:

- линейное отображение;
- расслоенное отображение.

Все мета-данные о логических томах и группах томов хранятся в области дескрипторов группы томов (*VGDA — Volume Group Descriptor Area*, которая работает аналогично таблице разделов. *VGDA* содержит:

- один дескриптор *PV*;
- один дескриптор *VG*;
- дескрипторы *LV*;
- несколько дескрипторов *PE*.

При старте операционной системы активируются логические тома и группы томов, а *VGDA* загружается в память. *VGDA* позволяет определить, где расположены *LV*. При попытке системы обратиться к устройству используется таблица соответствия (при помощи *VGDA*) для определения физического адреса, используемого в операции ввода-вывода.

Система *LVM* позволяет делать снимки («снапшоты»). Они позволяют создавать новые блочные устройства с точной копией логического тома, «замороженного» в какой-то момент времени. Обычно это используется в пакетных режимах. Например, при создании резервной копии системы. Однако при этом вам не будет нужно останавливать работающие задачи, меняющие данные на файловой системе. Когда необходимые процедуры будут выполнены, системный администратор может просто удалить устройство-«снапшот».

2. Создание дисков *LVM*. Для работы с *LVM* будем использовать утилиту *lvm2*. **Все действия необходимо выполнять от имени суперпользователя.** Согласно модели представления дискового пространства *LVM* для создания логических томов, доступных программам, необходимо выполнить ряд этапов.

Первый этап - инициализация дисков или разделов. Первым делом необходимо определить диски и/или логические разделы дисков, которые будем объединять в группы томов. Для этого создадим два фиксированных виртуальных диска и подключим их к контроллеру виртуальной машины. Посмотрим их буквенно-цифровое обозначение (две последние строки на рис. 2).

```
main@main-VirtualBox:~/Рабочий стол$ ls -l /dev | grep sd
brw-rw---- 1 root disk      8,  0  anp 11 17:48 sda
brw-rw---- 1 root disk      8,  1  anp 11 17:48 sda1
brw-rw---- 1 root disk      8,  2  anp 11 17:48 sda2
brw-rw---- 1 root disk      8,  3  anp 11 17:48 sda3
brw-rw---- 1 root disk     8, 16  anp 11 17:48 sdb
brw-rw---- 1 root disk     8, 32  anp 11 17:48 sdc
```

Рис. 2. Буквенно-цифровое обозначение дисков

Чтобы понять, какие файлы соответствуют подключенным дискам, можно поступить следующим образом. До подключения дисков к виртуальной машине выполнить команду *ls — l /dev | grep sd*. Затем подключить диски к виртуальной машине и вновь выполнить эту

команду. Появятся два новых файла — они и будут соответствовать подключенным дискам.

Инициализация дисков осуществляется командой `pvccreate /dev sdb /dev/sdc` (рис. 3).

```
root@main-VirtualBox:~# pvccreate /dev/sdb /dev/sdc
Physical volume "/dev/sdb" successfully created
Physical volume "/dev/sdc" successfully created
```

Рис. 3. Инициализация дисков

После выполнения этой команды в начале каждого диска создастся область дескрипторов группы томов.

Операции с физическими томами показаны в таблице.

| Команда [-опции] | Действие |
|--|---|
| <code>pvccreate <диск_1(раздел_1)> [<диск_2(раздел_2)> <...>]</code> | Инициализация дисков (разделов). В качестве аргументов используется буквенно-цифровое обозначение дисков (разделов) |
| <code>pvsscan</code> | Сканирование и вывод на экран всех инициализированных <i>LVM</i> дисков. Опции задают различный формат и количество выводимых данных |
| <code>pvdisplay</code> | Вывод на экран подробной информации о каждом <i>LVM</i> физическом диске. Опции задают различный формат и количество выводимых данных |
| <code>pvremove <диск(раздел)></code> | Удаление физического тома. Перед удалением том должен быть демонтирован |

Второй этап — создание группы томов. После создания физических томов можно создавать группу (группы) томов. На данном этапе создаются логические и физические экстенды. Количество физических экстендов для каждого физического диска определяется его размером и размером экстенда. По умолчанию размер экстенда составляет 4 Мбайт.

Основные операции с группами томов показаны в таблице.

| Команда [-опции] | Действие |
|--|---|
| <i>vgcreate</i> <VG_name> </dev/sdb> </dev/sdc> <...> | Создание группы томов, в которую входят диски /dev/sdb и /dev/sdc. VG_name — имя группы томов |
| <i>vgdisplay</i> | Вывод информации о группах томов |
| <i>vgreduce</i> <VG_name> <PV_name> | Удаление физического тома из группы томов. PV_name — имя физического тома |
| <i>vgextend</i> <VG_name> <PV_name> | Добавление нового физического тома в группу томов. Предварительно добавляемый том должен быть инициализирован |
| <i>vgremove</i> <VG_name> | Удаление группы томов |

Для удаления физического тома из группы томов необходимо освободить на нем все занятые физические экстененты. Это делается путем перераспределения занятых физических экстенентов на другие физические тома.

Для удаления группы томов ее необходимо деактивировать командой *vgchange -a n <name_VG>*.

Третий этап - создание логических томов. Каждая группа томов может быть разбита на логические тома.

Операции создания и удаления логических томов показаны в таблице.

| Команда [-опции] | Действие |
|--|---|
| <i>lvcreate -l 50 -n lv_01 vg_01</i> | Создание логического тома размером в 50 экстенентов с именем lv_01 на vg_01 |
| <i>lvcreate -L 500M -n lv_01 vg_01</i> | Создание логического тома размером 500 Мбайт с именем lv_01 на vg_01 |
| <i>lvscan</i> | Сканирование и вывод на экран всех инициализированных LVM логических томов |
| <i>lvdisplay</i> | Вывод на экран подробной |

| | |
|---|--|
| | информации о каждом <i>LVM</i> логическом томе |
| <i>lvremove <VG_name/LV_name></i> | Удаление логического тома из группы томов |

3. Утилиты работы с дисками. Рассмотрим работу некоторых утилит, необходимых для редактирования разделов дисков, создания на них файловых систем и проверки файловых систем на наличие ошибок. Их необходимо запускать с правами суперпользователя.

Утилита *fdisk* позволяет проводить операции с таблицами разделов диска типов *GPT*, *MBR*, *Sun*, *SGI* и *BSD*. Возможны два режима работы утилиты: обычный (конкретное действие задается опцией) и интерактивный. Для входа в интерактивный режим необходимо выполнить команду *fdisk <disk>* без указания опций. Все изменения будут выполняться в оперативной памяти до тех пор, пока не будет выполнен выход из интерактивного режима с сохранением изменений. Ввод в командную строку кода «*m*» позволит вывести список всех операций, которые способна выполнить данная утилита. Основные операции интерактивного режима приведены в таблице.

| Код | Операция |
|----------------|---|
| <i>F</i> | Вывод информации о неразмеченной области диска |
| <i>n</i> | Создание нового раздела диска |
| <i>d</i> | Удаление раздела диска |
| <i>p</i> | Вывод на экран таблицы разделов |
| <i>g/G/o/s</i> | Создание новой таблицы разделов: <i>g</i> — <i>GPT</i> , <i>G</i> — <i>SGI</i> , <i>o</i> — <i>DOS</i> , <i>s</i> — <i>Sun</i> . При этом произойдет удаление предыдущей таблицы разделов |

В обычном режиме полезной оказывается опция, показанная в таблице.

| Команда [-опции] | Действие |
|-------------------|--|
| <i>fdisk -l</i> | Вывод информации о всех подключенных к компьютеру дисках (блочных устройствах). При этом отображаются также диски без файловой системы, несмонтированные и диски <i>LVM</i> |

Утилита *mkfs* позволяет наносить различные файловые системы на размеченные диски (обычные и *LVM*). В таблице приведены некоторые варианты ее использования.

| Команда [-опции] | Действие |
|--|---|
| <i>mkfs -t ext3 <диск (раздел)></i> | Создание файловой системы <i>ext3</i> на указанном диске (разделе диска). Аргумент <i><диск (раздел)></i> - это буквенно-цифровое обозначение диска (раздела) |
| <i>mkfs -t btrfs <диск (раздел)></i> | Создание файловой системы <i>btrfs</i> на указанном диске (разделе диска) |

Утилита *blkid* выводит на экран информацию об атрибутах блочных устройств. В качестве атрибутов могут выступать метка (*label*) и уникальный идентификатор (*UUID*) блочного устройства (диска, раздела диска).

Утилита *lsblk* позволяет получить список блочных устройств в иерархическом виде. Пример работы данной утилиты показан на рис. 4. Информация о файловой системе отображается по столбцам.

```

root@main-VirtualBox:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                 8:0    0   16G  0 disk
├─sda1                             8:1    0    9,3G  0 part /
├─sda2                             8:2    0   977M  0 part [SWAP]
└─sda3                             8:3    0    5,7G  0 part /home
sdb                                 8:16   0    500M  0 disk
sdc                                 8:32   0     1G  0 disk
└─vg_01-lv_02                     252:1   0     1G  0 lvm
sdd                                 8:48   0    1,5G  0 disk
├─sdd1                             8:49   0    500M  0 part
│   └─vg_01-lv_01                 252:0   0    400M  0 lvm
│       └─vg_01-lv_02             252:1   0     1G  0 lvm
│           └─vg_01-lv_03         252:2   0    92M  0 lvm
└─sdd2                             8:50   0    500M  0 part
    └─sdd3                         8:51   0    500M  0 part

```

Рис. 4. Иерархия дисков

Примеры использования утилит *blkid*, *lsblk* приведены в таблице.

| Команда [-опции] | Действие |
|------------------------------------|--|
| <i>blkid <диск (раздел)></i> | Вывод на экран <i>UUID</i> диска или раздела диска |
| <i>lsblk</i> | Вывод на экран иерархии дисков |

| | |
|-----------------|--|
| <i>lsblk -b</i> | Вывод на экран иерархии дисков. Размер дисков отображается в байтах |
| <i>lsblk -f</i> | Вывод на экран информации в столбцах NAME, FSTYPE, LABEL, UUID, MOUNTPOINT |
| <i>lsblk -e</i> | Вывод на экран назначения каждого столбца |

Утилита *fsck* позволяет провести проверку и восстановление файловой системы диска или раздела диска. В процессе работы данные не сразу записываются на диск, а некоторое время находятся в оперативной памяти и при неожиданном выключении файловая система может быть повреждена. До начала проверки файловой системы ее необходимо демонтировать. Работу утилиты *fsck* изучить самостоятельно.

ЗАДАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ

1. Изучить теоретический материал.
2. Открыть менеджер виртуальных машин и создать три виртуальных жестких диска фиксированного типа размерами 500 Мб, 1 Гб и 1,5 Гб соответственно. Подключить их к контроллеру виртуальной машины.
3. Загрузить ОС Linux и авторизоваться в консоли № 1.
4. Определить буквенно-цифровое обозначение трех подключенных дисков и выполнить инициализацию первых двух дисков. Результат инициализации проверить утилитой *pvdisk*.
5. Создать группу томов *vg_01*, включающую два физических диска объемом 500 Мб и 1 Гб. Результат проверить утилитой *vgdisplay*.
6. Разбить группу томов *vg_01* на три логических тома *lv_01*, *lv_02*, *lv_03* согласно таблице.

| LV name | Количество экстенгов | Размер логического тома | Алгоритм отображения экстенгов |
|--------------|----------------------|-------------------------|--------------------------------|
| <i>lv_01</i> | 100 | — | Линейный |
| <i>lv_02</i> | — | 1Гб | Линейный |
| <i>lv_03</i> | 23 | — | Линейный |

7. Создать на диске объемом 1,5 Гб таблицу разделов *GPT*, которая включает **три** раздела равного объема.
 - 7.1. Инициализировать полученные разделы.
 - 7.2. Создать группу томов *vg_02*, состоящую из инициализированных выше трех разделов.
 - 7.3. Разбить группу томов *vg_02* на **пять** логических томов так, чтобы все физические экстенды группы были задействованы.
 - 7.4. Результаты проверить соответствующими утилитами.
8. Заполнить таблицу для любых двух физических томов, принадлежащих разным группам томов.

| PV name | VG name | PV size | PE size | Total PE | Free PE |
|---------|---------|---------|---------|----------|---------|
| | | | | | |
| | | | | | |

9. Определить буквено-цифровые обозначения полученных в п. 6 и п. 7 логических дисков *LVM* (их всего 8).
10. Нанести на них файловые системы, а затем смонтировать в точки монтирования согласно таблице.

| LV name | VG name | Точка монтирования | Тип файловой системы |
|--------------|--------------|--------------------|----------------------|
| <i>lv_01</i> | <i>vg_01</i> | <i>~/disc_1_1</i> | <i>ext2</i> |
| <i>lv_02</i> | | <i>~/disc_1_2</i> | <i>ext3</i> |
| <i>lv_03</i> | | <i>~/disc_1_3</i> | <i>ext4</i> |
| <i>lv_01</i> | <i>vg_02</i> | <i>~/disc_2_1</i> | <i>btrfs</i> |
| <i>lv_02</i> | | <i>~/disc_2_2</i> | <i>xfs</i> |
| <i>lv_03</i> | | <i>~/disc_2_3</i> | <i>jfs</i> |
| <i>lv_04</i> | | <i>~/disc_2_4</i> | <i>zfs</i> |
| <i>lv_05</i> | | <i>~/disc_2_5</i> | <i>reiserfs</i> |

11. Определить номера физических и логических экстендов группы томов *vg_01*.
12. Определить, какие физические экстенды соответствуют логическим экстендам для группы томов *vg_01* и на каких физических дисках они расположены.
13. Ознакомиться с работой утилит *blkid*, *lsblk* и *fsck*.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Oracle VM VirtualBox, User manual. – Version 5.1.26. – 2017.
2. Забродин Л.Д. UNIX: основы командного интерфейса и программирования (в примерах и задачах) / Л.Д. Забродин, В.В. Макаров, А.Б. Вавренюк, – М.: Национальный исследовательский ядерный университет «МИФИ», 2010.
3. Костромин В.А. Linux для пользователя / В.А. Костромин. – СПб.: БХВ-Петербург, 2003.
4. Кристофер Негус. Ubuntu и Debian Linux для продвинутых: более 1000 незаменимых команд / Кристофер Негус, Франсуа Каэн. – СПб.: Питер, 2011.

Работа с файловой системой LINUX

Составители: Бубнов Сергей Алексеевич
Бубнов Алексей Алексеевич

Редактор Р.К. Мангутова

Корректор С.В. Макушина

Подписано в печать 27.06.19. Формат бумаги 60х84 1/16.

Бумага писчая. Печать трафаретная. Усл. печ. л. 1,75

Тираж 50 экз. Заказ

Рязанский государственный радиотехнический университет.

390005, Рязань, ул. Гагарина, 59/1.

Редакционно-издательский центр РГРТУ.