

ЛАБОРАТОРНАЯ РАБОТА № 7

LVM

Цель работы – Научится создавать и обслуживать разделы LVM.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1 Управление файловыми системами

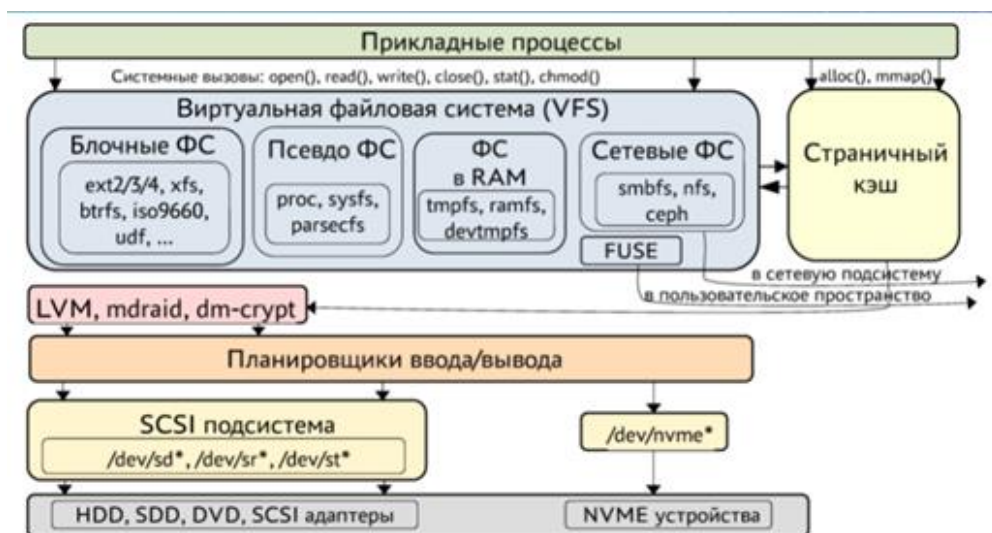


Рисунок 1 – Архитектура подсистемы хранения данных

Именованье файлов дисковых устройств:

- `/dev/sd*` - дисковые устройства с интерфейсом последовательной передачи данных (SCSI, SATA, см. man 4 sd)
- `/dev/hd*` дисковые устройства с интерфейсом параллельной передачи данных (PATA, см. man 4 hd)
- Буква (a,b,c,...) после sd или hd – номер диска
- Число после буквы – номер раздела (`/dev/sdb4`)
- Если числа нет, то весь диск (`/dev/sdc`)
- В случае BIOS/MBR: поддерживается 4 первичных раздела (1-4), один из которых может объявлен, как расширенный. В расширенном разделе могут быть созданы логические разделы (первый логический раздел -5)
- В UEFI/GPT все разделы первичные (1-128)

Поддерживаемые типы файлов системы:

- модули ядра (драйверы) поддерживаемых файловых систем в установленной операционной системы `ls/lib/modules/$(uname -r)/kernel/fs`
- список драйверов файловых систем, загруженных в данный момент `cat/proc/filesystems` (nodev-псевдофайловые или временные файловые системы)
- все ФС должны предоставлять VFS информацию о суперблоке, inode, dentry, блоках данных
- «родные» для Linux ФС – ext2/ext3, ext4

- дисковые ФС, поддерживаемые ядром: XFS, BtrFS, iso9660, udf.

Файловая система ext2.

- Макс. размер файла:
 - 16 GiB (размер блока 1 KiB)
 - 1 TiB (размер блока 4 KiB)
- Макс. размер ФС:
 - 4 TiB (размер блока 1 KiB)
 - 16 TiB (размер блока 4 KiB)
- Пространство ФС разбито на группы блоков: резервный суперблок, таблица описания группы, битовые карты inode и блоков данных, области inode и блоков данных (dumpe2fs)

Суперблок ФС содержит информацию о самой файловой системе (тип, размер, состояние, UUID и т.д.). Содержимое суперблока можно посмотреть командой: `tune2fs -l/dev/имя_раздела`

Суперблок в ФС дублируется:

`dumpe2fs/dev/имя_раздела | grep superblock` позволяет определить номера резервных суперблоков.

При монтировании устройства можно указать резервный суперблок (параметр `-o sb=n` команды `mount`, n- номер резервного суперблока).

Размер inode хранится в суперблоке (128.256 байт).

Начиная с ядра 2.6.10 – 256 байт (для хранения точного времени и расширенных атрибутов, включая метку безопасности (мандатную метку)). Команда для просмотра inode: `debugfs -R «stat имя_файла» имя_блочного_устройства`

Для адресов блоков данных отведено: 12 прямых указателей, 1 косвенный указатель, 1 – указатель с двойной косвенной адресацией, 1 – с тройной косвенной адресацией.

Файловая система ext3. Главные отличия от ext2:

- наличие журнала,
- онлайн увеличение размера ФС,
- использование сбалансированного дерева для индексирования больших каталогов обеспечило более быстрый поиск файлов,
- ext3 совместима с ext2 (ext2 может быть преобразована в ext3 (`tune2fs -j`) и наоборот (`tune2fs -O^has_journal`),

- режимы работы журнала: `journal`, `ordered`, `writeback`,
- журнал обычно размещается в конце раздела ФС,
- максимальный размер ФС 1 EiB, файла -16 Тб (при размере блока 4 Кб).

Отличительные черты:

- размещение данных экстендами,
- отложенное распределение (выделение) блоков – распределение блоков откладывается до тех пор, пока не пойдет запись на диск,
- предварительное выделение места для файла на диске,
- контрольные суммы журнала – для определения возможных проблем в журнале.

Основные характеристики ФС XFS:

-64-х разрядная ФС (максимальный размер файла -8EiB, максимальный размер ФС 8EiB)

- увеличение производительности путем использования линейных областей (allocation groups)

- использование экстендов для выделения места в области данных

- журналируемая ФС (только метаданные)

- индексные дескрипторы выделяются динамически

- поддержка дефрагментации «на лету»

- возможность увеличения размера ФС «на лету»

- поддержка отложенного выделения места (delayed allocation).

Основные характеристики ФС Btrfs:

- 64х- разрядная ФС (макс. Размер файла 16 EiB, раздела – 16 EiB)

- поддержка механизма «кэширования при записи» (Copy On Write, COW)

- поддержка подтомов (subvolumes)

- поддержка снимков состояния ФС (использует механизм COW и подтома)

- дефрагментация и сжатие данных «на лету»

- динамическое размещение inode

- целостность данных (вычисление контрольных сумм для данных и метаданных)

- встроенная поддержка многодисковых ФС (RAID и LVM)

2 Создание дисковых разделов

Утилиты для разметки диска:

fdisk имя_диска

parted имя_диска

GUID Partition Table, GPT — стандарт формата размещения таблиц разделов на физическом жестком диске используется в настоящее время вместо MBR.

Утилиты создания разделов:

- **sfdisk** - предназначена для использования в сценариях (скриптах)

- **cdisk** - псевдографическая утилита

- **gparted** - графическая утилита

Список дисков и разделов **fdisk -l**

Список разделов на одном диске **fdisk -l имя_диска**

Запуск **fdisk** в интерактивном режиме **fdisk имя_диска**

Например, **sudo fdisk /dev/sdb**

Основные команды **fdisk**:

m - помощь

a - включение или выключения флага boot для раздела;

d - удалить раздел;

F - показать свободное место;

l - вывести список известных типов разделов;

n - создать новый раздел;

p - вывести таблицу разделов;

t - изменение типа раздела;
i - вывести информацию о разделе;
I и O - записать или загрузить разметку в файл сценария sfdisk;
w - записать новую таблицу разделов на диск;
q - выйти без сохранения;
g - создать пустую таблицу разделов GPT;
o - создать пустую таблицу разделов MBR.

Примечание. Если после записи таблицы разделов fdisk команда lsblk не показывает созданные разделы, то надо вызвать команду partprobe, чтобы уведомить ядро об изменении таблицы разделов.

Основные команды **parted**:

help команда - помощь по выбранной команде;

mkpart тип раздела файловая система начало конец - создание раздела linux с файловой системой начиная с позиции начало заканчивая конец, два последних параметра задаются в мегабайтах по умолчанию;

mktable тип - создать таблицу разделов;

print - отобразить таблицу разделов;

quit - выйти;

resizepart раздел конец - изменить размер раздела;

rm раздел - удалить раздел;

select раздел - установить раздел как текущий;

set раздел флаг состояние - установить флаг для раздела. Состояние может быть on (включен) или off(выключен);

unit единицы – установка единиц измерения (s, MiB, GiB, MB, GB)

Утилиту parted можно использовать в командной строке

Parted имя_диска команды

3 Создание файловой системы

ФС создается командой mkfs.тип_ФС параметры файл_устройства

Количество индексных дескрипторов для файловых систем семейства ext задается при создании ФС и фиксированно.

Параметр -i задает плотность индексных дескрипторов

Установки, которые применяются при создании ФС по умолчанию, находятся в etc/mke2fs.conf

Чтобы отмонтированный раздел диска был доступен, нужно подключить его к каталогу в дереве ФС (точке монтирования).

Варианты монтирования ФС:

- временное монтирование с помощью mount
- постоянное монтирование с помощью etc/fstab
- монтирование с помощью systemd

Временное	монтирование.	Команда	sudo	mount/dev/устройство
точка_монтирования				

Точка монтирования должна быть предварительно создана – обычно это пустой каталог (можно использовать каталог /mnt)

Для размонтирования используется один из вариантов:

`sudo umount/dev/устройство`

или

`sudo umount точка_монтирования`

Примечание. Чтобы размонтирование прошло успешно, на устройстве не должно быть занятых файловых ресурсов.

Настройка автоматического монтирования ФС может быть выполнена путем соответствующих настроек в /etc/fstab

Файл /etc/fstab содержит следующие поля:

- файл устройства/метка/UUID
- точка монтирования
- тип файловой системы
- параметры (обычно defaults)
- признак для команды dump (обычно 0)
- признак для команды fsck (1 – для корневой файловой системы, 2 – для остальных)

Для монтирования ФС через **systemd** требуется создать юнит типа mount, в котором следует описать какое устройство и как должно быть смонтировано. Название юнита должно совпадать с именем точки монтирования, но вместо символа «/» должен использоваться символ «-».

В юните должна быть секция [mount] со следующими параметрами:

- What - имя устройства (имя файла устройства, метка, UUID)
- Where - точка монтирования
- Type - тип ФС
- Options - параметры монтирования

Утилиты для работы с файловой системой:

- tune2fs - настройка параметров ФС
- dumpe2fs - вывод информации о структуре ФС
- e2fsck – проверка целостности структуры ФС
- resize2fs – изменение размеров ФС
- e4defrag – дефрагментация ФС ext4
- debugfs - отладчик ФС
- e2image - сохранения метаданных ФС в файл
- df (-h, -i) – информация о свободном месте в областях данных и inode
- du -sh каталог – общий размер файлов в каталоге
- lsblk – список блочных устройств

4 Управление логическими томами LVM

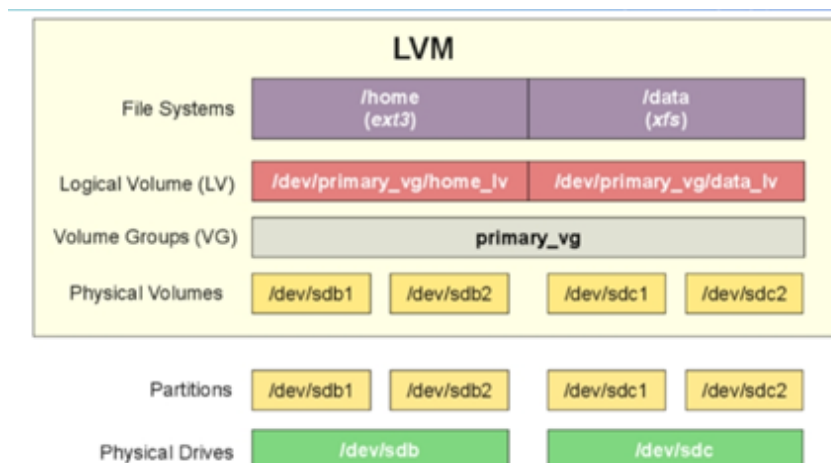


Рисунок 2 – LVM

Работа с томами с помощью LVM происходит на 3-х уровнях абстракции:

1. Физический уровень (PV). Сначала диск инициализируется командой **pvcreate** — в начале диска создается дескриптор группы томов. При этом важно заметить, что диск не обязательно должен быть физическим — мы можем отметить на использование обычный раздел диска.

2. Группа томов (VG). С помощью команды **vgcreate** создается группа томов из инициализированных на предыдущем этапе дисков.

3. Логический том (LV). Группы томов нарезаются на логические тома командой **lvcreate**.

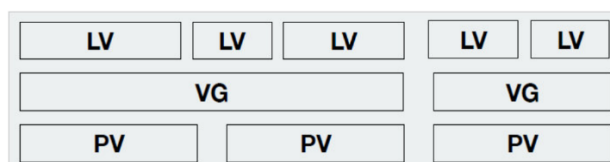


Рисунок 3 – Уровни абстракции

Вывод списка блочных устройств:

```
administrator@rator:~$ sudo lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   8G  0 disk
└─sda1       8:1    0   8G  0 part /
sdb          8:16    0   8G  0 disk
sdc          8:32    0   8G  0 disk
sdd          8:48    0   8G  0 disk
sde          8:64    0   8G  0 disk
sr0         11:0    1  8.6G  0 rom
```

Установка пакета LVM – `sudo apt install lvm2 -y`

Создание физических томов с помощью `pvcreate /dev/sdb`

Посмотреть, что диск может использоваться LVM можно командой:

`pvdisplay` – вывод атрибутов PV, `pvscan` – сканирование дисков на PV, `pvs` – вывод информации о PV.

```

administrator@grator:~$ sudo pvdisplay
"/dev/sdc" is a new physical volume of "8,00 GiB"
--- NEW Physical volume ---
PV Name                /dev/sdc
VG Name
PV Size                8,00 GiB
Allocatable            NO
PE Size                0
Total PE               0
Free PE                0
Allocated PE           0
PV UUID                8smd8A-Hdf8-YLdS-Hwz0-DenI-Hxtq-AdLCqd

"/dev/sdb" is a new physical volume of "8,00 GiB"
--- NEW Physical volume ---
PV Name                /dev/sdb

```

PV Name — имя диска.

VG Name — группа томов, в которую входит данный диск (в нашем случае пусто, так как мы еще не добавили его в группу).

PV Size — размер диска.

Allocatable — распределение по группам. Если NO, то диск еще не задействован и его необходимо для использования включить в группу.

PE Size — размер физического фрагмента (экстента). Пока диск не добавлен в группу, значение будет 0.

Total PE — количество физических экстенгов.

Free PE — количество свободных физических экстенгов.

Allocated PE — распределенные экстенги.

PV UUID — идентификатор физического раздела.

Создание групп томов

Инициализированные на первом этапе диски должны быть объединены в группы.

Группа может быть создана:

```
vgcreate vg01 /dev/sdb /dev/sdc
```

vg01 — произвольное имя создаваемой группы; /dev/sdb, /dev/sdc — наши диски

Просмотреть информацию о созданных группах можно командой:

```
vgdisplay
```

VG Name — имя группы.

Format — версия подсистемы, используемая для создания группы.

Metadata Areas — область размещения метаданных. Увеличивается на единицу с созданием каждой группы.

VG Access — уровень доступа к группе томов.

VG Size — суммарный объем всех дисков, которые входят в группу.

PE Size — размер физического фрагмента (экстента).

Total PE — количество физических экстенгов.

Alloc PE / Size — распределенное пространство: количество экстенгов / объем.

Free PE / Size — свободное пространство: количество экстенгов / объем.

VG UUID — идентификатор группы.

Создание логических томов

Последний этап — создание логического раздела из группы томов командой `lvcreate`. Ее синтаксис: `lvcreate [опции] <имя группы томов>`

Примеры создания логических томов:


```
administrator@rator:~$ sudo lvcreate -L 9G -n lv01 vg01
Logical volume "lv01" created.
```

Информация о логических томах `lvdisplay`, `lvscan`, `lvs`:

LV Path — путь к устройству логического тома.

LV Name — имя логического тома.

VG Name — имя группы томов.

LV UUID — идентификатор.

LV Write Access — уровень доступа.

LV Creation host, time — имя компьютера и дата, когда был создан том.

LV Size — объем дискового пространства, доступный для использования.

Current LE — количество логических экстендов.

Создание файловой системы и монтирование тома

Чтобы начать использовать созданный том, необходимо его отформатировать, создав файловую систему и примонтировать раздел в каталог.

Процесс создания файловой системы на томах LVM ничем не отличается от работы с любыми другими разделами.

Например, для создания файловой системы `ext4`:

```
mkfs.ext4 /dev/vg01/lv01
```

vg01 — наша группа томов; **lv01** — логический том.

Далее надо создать точку монтирования и смонтировать том

```
administrator@rator:~$ sudo mkdir /data
administrator@rator:~$ sudo mount /dev/vg01/lv01 /data
```

где `/dev/vg01/lv01` — созданный нами логический том, `/data` — раздел, в который мы хотим примонтировать раздел.

Создаем юнит `systemd` для монтирования

```
administrator@rator:~$ sudo vi /etc/systemd/system/data.mount
```

```
[Unit]
Description=Mount data

[Mount]
What=/dev/vg01/lv01
Where=/data
Type=ext4
Option=defaults,noexec

[Install]
WantedBy=multi-user.target
```

Проверяем файловую систему на наличие ошибок с помощью `fsck.ext4`

```
administrator@rator:~$ sudo fsck.ext4 /dev/vg01/lv01
e2fsck 1.43.4 (31-Jan-2017)
/dev/vg01/lv01: clean, 11/589824 files, 62641/2359296 blocks
```

или `e2fsck`

```
administrator@rator:~$ sudo e2fsck -f /dev/vg01/lv01
e2fsck 1.43.4 (31-Jan-2017)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/vg01/lv01: 11/589824 files (0.0% non-contiguous), 62641/2359296 blocks
```

Чтобы уменьшить размер логического тома, сначала необходимо уменьшить размер файловой системы и затем том с помощью команды `resize2fs`


```

administrator@rator:~$ sudo resize2fs -p /dev/vg01/lv01 7G
resize2fs 1.43.4 (31-Jan-2017)
Resizing the filesystem on /dev/vg01/lv01 to 1835008 (4k) blocks.
Begin pass 3 (max = 72)
Scanning inode table      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
The filesystem on /dev/vg01/lv01 is now 1835008 (4k) blocks long.

```

Затем уменьшить размер логического тома с 9 Гб до 7Гб
 lvreduce -L 7G /dev/vg01/lv01

Удаление физического тома

Перед удалением физического тома надо переместить экстенды с него

```

administrator@rator:~$ sudo pvmove /dev/sdb
/dev/sdb: Moved: 0,56%
/dev/sdb: Moved: 100,00%

```

Командой vgreduce физический том удалить из группы томов:

```

administrator@rator:~$ sudo vgreduce vg01 /dev/sdb
Removed "/dev/sdb" from volume group "vg01"

```

Удалить сам физический том:

```

administrator@rator:~$ sudo pvremove /dev/sdb
Labels on physical volume "/dev/sdb" successfully wiped.

```

Добавление физического тома

Для этого необходимо создать физический том suda pvcreate /dev/sdb.

Добавить /dev/sdb в группу физических томов sudo vgextend vg01 /dev/sdb

Увеличим размер логического тома на 3Гб

sudo lvextend -l +3G dev/vg01/lv01

Увеличиваем файловую систему

sudo resize2fs dev/vg01/lv01

Работа со снимками

Снимки диска позволят нам откатить состояние на определенный момент. Это может послужить быстрым вариантом резервного копирования. Однако нужно понимать, что данные хранятся на одном и том же физическом носителе, а значит, данный способ не является полноценным резервным копированием.

Необходимо смонтировать файловую систему и создать произвольный файл, по которому можно изучить снимки состояний.

Создать снимок можно: sudo systemctl start data.mount

```

administrator@rator:~$ sudo systemctl start data.mount
1 /dev/vg01/lv01
Using default stripesize 64,00 KiB.
Logical volume "snap090721" created.

```

Затем надо создать каталог и смонтировать снимок состояния

```

administrator@rator:~$ sudo mkdir /snap01
administrator@rator:~$ sudo mount /dev/vg01/snap090721 /snap01

```

МЕТОДИКА ВЫПОЛНЕНИЯ

1 Работа с жесткими дисками

1. Используйте `fdisk -l`, чтобы посмотреть информации о дисках и разделах на вашем компьютере. Какие из ваших дисков в системе имеют разметку? Описать. Посмотрите свободное место на этих разделах с помощью команды `df`

2. Добавьте 2 новых жестких диска в виртуальную машину (SCSI и IDE). Для добавления новых дисков виртуальную машину нужно выключить. После добавления новых дисков загрузите Astra Linux. Далее проверьте, что новые диски появились в системе.

Например, один из наших новых дисков определился в системе как `/dev/sdc`.

Используя команду `fdisk`, создайте два новых раздела (`/dev/sdc1` и `/dev/sdc2`) на данном диске, размером 512 М. Перезагрузитесь, чтобы удостовериться, что вы корректно изменили разделы на жестком диске.

3. Используя команду `mke2fs` или команду `mkfs.ext2`, создайте новую файловую систему `ext2` на новом логическом разделе `/dev/sdc1`.

4. На разделе `/dev/sdc2` создайте файловую систему `ext4`

5. Создайте директорию `/data`, в которую вы будете монтировать новый логический том `/dev/sdc2`.

6. Используйте команду `mount`, чтобы смонтировать новый логический том в директорию `/data`. Скопируйте `/etc/passwd` в директорию `/data` и проверьте, что копирование было успешно.

7. Затем размонтируйте директорию. Еще раз проверьте содержимое каталога `/data`.

8. Добавьте метку `/data` к новому разделу с помощью команды `e2label`.

9. Отредактируйте файл `/etc/fstab` так, чтобы новый раздел монтировался при загрузке системы. Проверьте, что вы правильно прописали данную строку в файл `fstab`

10. Перезагрузите систему и убедитесь, что новый раздел монтируется автоматически.

11. Посмотрите свойства файловой системы для раздела `/dev/sdc1`. Включено ли журналирование данной файловой системы? Если нет, то включите журналирование для раздела `/dev/sdc1`. Создайте для данного раздела точку монтирования `/data1` и смонтируйте туда этот слайс.

12. Перейдите в раздел `/data1` и попробуйте размонтировать его. Почему вы не можете размонтировать данный раздел?

13. Посмотрите, кто из пользователей, и какими процессами занял раздел `/data1`. Завершите все процессы на разделе `/data1`

14. Попробуйте размонтировать раздел `/data1` снова. Что произошло?

15. Увеличьте резервируемое место (`minfree`) файловой системой до 10% на разделе `/dev/sdc1`. Осуществите проверку файловой системы для раздела `/dev/sdc1`

2 LVM

Для данного раздела нужно установить пакет `lvmm2` (если он у вас не установлен).

1. Создайте три новых раздела на диске по 512 М. При необходимости добавьте в систему новый диск (если вы работаете на виртуальной машине). Предположим, что в нашем случае мы добавили новый диск и он был распознан системой как `/dev/sda`

(соответственно, новые разделы будут /dev/sda1, /dev/sda2 и /dev/sda3). Покажите и опишите свой диск и разделы. Разделы на диске должны иметь тип LVM

2.LVM строится на основе разделов жёсткого диска и/или целых жёстких дисков. Поэтому на первых двух из созданных нами разделов (/dev/sda1, /dev/sda2) создайте физический том (physical volume). Проверьте, что физические тома созданы корректно.

3.На первых двух физических томах (/dev/sda1 и /dev/sda2) создаём группу томов, которая будет называться, например, vg1. Проверьте, что группа vg1 создана корректно. Групп можно создать несколько, каждая со своим набором томов. Но обычно это не требуется.

4.В группе томов создайте логический том lv1 размером 200 Мб и lv2 размером 300 Мб. Проверьте, что том создан корректно.

5.Теперь у вас есть блочные устройства /dev/vg1/lv1 и /dev/vg1/lv2 . Создайте на них файловую систему: на lv1 – ext4 и на lv2 – ext3.

Примечание. Удаление LVM (или отдельных его частей, например, логических томов или групп томов) происходит в обратном порядке - сначала нужно отмонтировать разделы, затем удалить логические тома (lvremove), после этого можно удалить группы томов (vgremove) и ненужные физические тома (pvremove). **На данном шаге удалять ничего не нужно!**

6.Смонтируйте созданные логические тома lv1 и lv2 в директории /lvdir1 и /lvdir2 (предварительно их создав). Посмотрите информацию по смонтированным томам, опишите. Выполните настройки для автоматического монтирования данных томов во время запуска системы. Для этого внесите изменения в файл /etc/fstab. Проверьте корректность правок в файле /etc/fstab без перезагрузки.

7.Чтобы добавить раздел /dev/sda3 в группу томов, создайте физический том для этого раздела. Далее добавьте его в группу.

7.Увеличьте размер существующего тома lv2 за счет физического тома, добавленного в группу. Для этого размонтируйте том lv2 и проверьте его на наличие ошибок. Увеличьте lv2 на 512М. После увеличения выполните проверку файловой системы на наличие ошибок. Подмонтируйте измененный том и проверьте, изменился ли его размер.

8.Чтобы убрать из работающей группы томов раздел /dev/sda2 сначала перенесите все данные с него на другие диски, затем удалите его из группы томов, после этого удалите физический том.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какой файл устройства соответствует 2-ому разделу 3-его SATA диска?
2. Какая команда помогает найти причину, почему не получается размонтировать ФС?
3. Какую утилиту следует использовать, чтобы создать на разделе ФС типа ext3?
4. В каком порядке следует уменьшать размер логического тома?
5. В каком порядке следует увеличивать размер логического тома?
6. Что следует сделать перед удалением физического тома?