

Практическая работа №10

Работа с томами в ОС Linux

Очень многие функциональные возможности в том или ином виде присутствуют во всех современных ОС. Аналоги возможностей «Управление дисками» и «Windows Storage Spaces» в операционных системах семейства Linux также присутствуют, хотя и выглядят по-другому, работают на немного других принципах и обладают несколько отличающимися возможностями. Данные компоненты называются mdadm (управление программными RAID-массивами) и LVM (управление логическими томами). Основное отличие от средств ОС Windows — они являются дополняющими друг друга, а не взаимоисключающими (в ОС Windows набор дисков может быть использован либо для создания томов в «Управлении дисками», либо для создания пространства хранения в Windows Storage Spaces).

Рассмотрим возможности этих средств на примере Astra Linux. Несмотря на наличие графических оболочек, лучшим вариантом будет работа через командную строку — такой вариант функциональнее, быстрее и информативнее. Разумеется, все возможности управления томами доступны только администратору, поэтому первой командой в терминале будет `sudo su` (или придётся дописывать `sudo` перед всеми последующими командами).

К сожалению, в Astra Linux данные компоненты относятся к дополнительным и не устанавливаются по-умолчанию. Исправим это досадное недоразумение командой

```
apt install mdadm lvm2
```

Согласимся с предлагаемыми изменениями.

К сожалению, в репозиториях Astra Linux отсутствует (согласно официальной документации) пакет `thin-provisioning-tools`, необходимый для работы с тонкими томами. Тем не менее, в случае нормальной работы этот пакет по факту не требуется, поэтому создадим «обманку», чтобы команды завершались успешно:

```
echo '#!/bin/sh' > /usr/sbin/thin_check
```

и команду

```
chmod +x /usr/sbin/thin_check
```

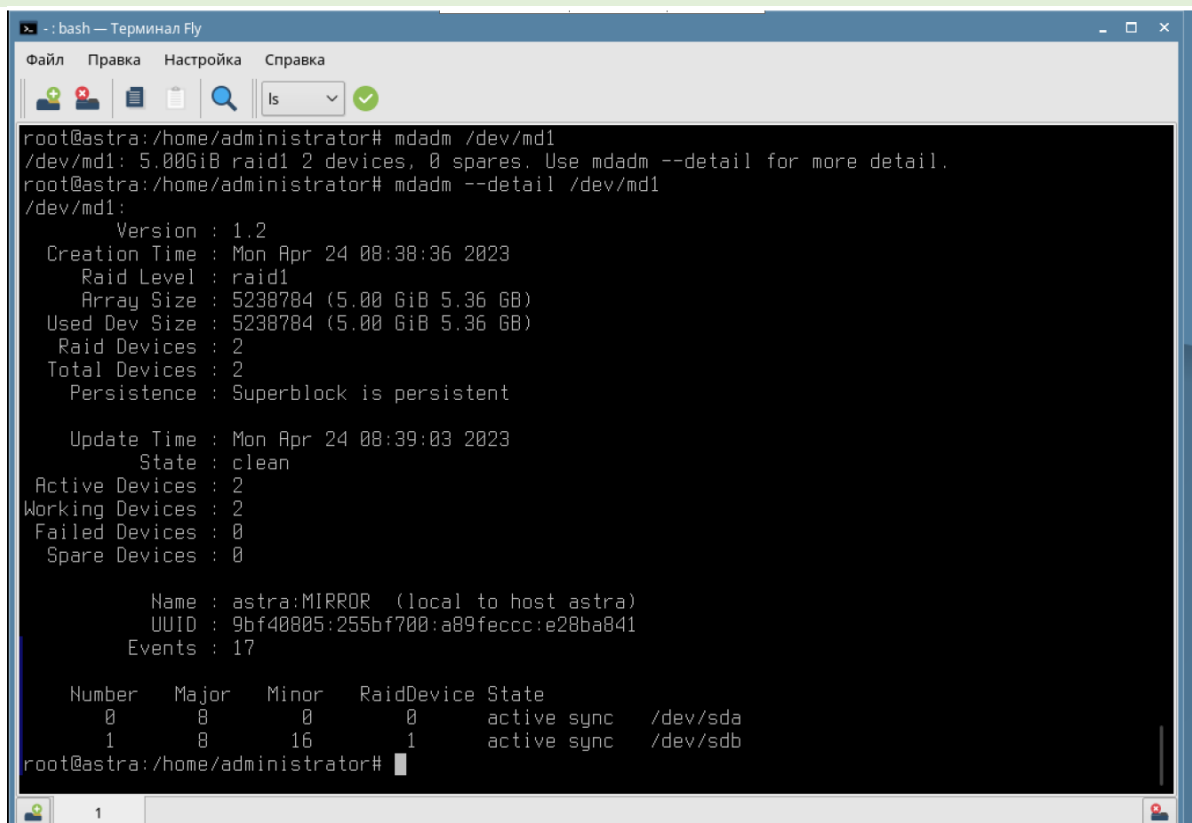
Для начала, попробуем создать зеркальный том. Как и в ОС Windows, данная возможность присутствует в обоих средствах, разница в том, что mdadm будет использовать целиком диск, LVM только указанный объём. Поскольку mdadm затратит на это несколько меньше системных ресурсов, мы выберем его. Это средство позволяет создавать и работать с программными массивами уровней 0,1,5,6,10 и некоторыми другими (JBOD также поддерживается). В данном случае, для работы томов с чётностью не требуется специальной редакции ОС.

Создадим зеркальный том командой

```
mdadm --create /dev/md127 --level 1 --name=MIRROR --raid-devices=2 /dev/sda /dev/sdb
```

Посмотрим параметры созданного массива командой

```
mdadm --detail /dev/md127
```



```
--: bash — Терминал Fly
Файл  Правка  Настройка  Справка
root@astra:/home/administrator# mdadm /dev/md1
/dev/md1: 5.00GiB raid1 2 devices, 0 spares. Use mdadm --detail for more detail.
root@astra:/home/administrator# mdadm --detail /dev/md1
/dev/md1:
  Version : 1.2
  Creation Time : Mon Apr 24 08:38:36 2023
  Raid Level : raid1
  Array Size : 5238784 (5.00 GiB 5.36 GB)
  Used Dev Size : 5238784 (5.00 GiB 5.36 GB)
  Raid Devices : 2
  Total Devices : 2
  Persistence : Superblock is persistent

  Update Time : Mon Apr 24 08:39:03 2023
  State : clean
  Active Devices : 2
  Working Devices : 2
  Failed Devices : 0
  Spare Devices : 0

  Name : astra:MIRROR (local to host astra)
  UUID : 9bf40805:255bf700:a89feccc:e28ba841
  Events : 17

   Number Major Minor RaidDevice State
    0       8       0        0     active sync  /dev/sda
    1       8      16        1     active sync  /dev/sdb
root@astra:/home/administrator#
```

Теперь для хранения данных на томе необходимо создать файловую систему. С данной командой мы уже сталкивались.

```
mkfs.ext4 /dev/md127
```

В отличие от ОС Windows, в ОС Linux используется другая концепция файловой системы. Поэтому, вторичные файловые системы не подключаются автоматически. Если во время загрузки ОС Windows обнаружит понятную ей файловую систему, она подключит её как дополнительный логический диск. В ОС Linux файловые системы монтируются в папки, поэтому автоматическое выполнение подобного действия невозможно, требуется в явном виде указать папку, в которую будет монтироваться файловая система. Для указания точек монтирования файловых систем служит файл настроек `/etc/fstab`

Давайте отредактируем этот файл и добавим туда точку монтирования файловой системы зеркального тома.

```
nano /etc/fstab
```

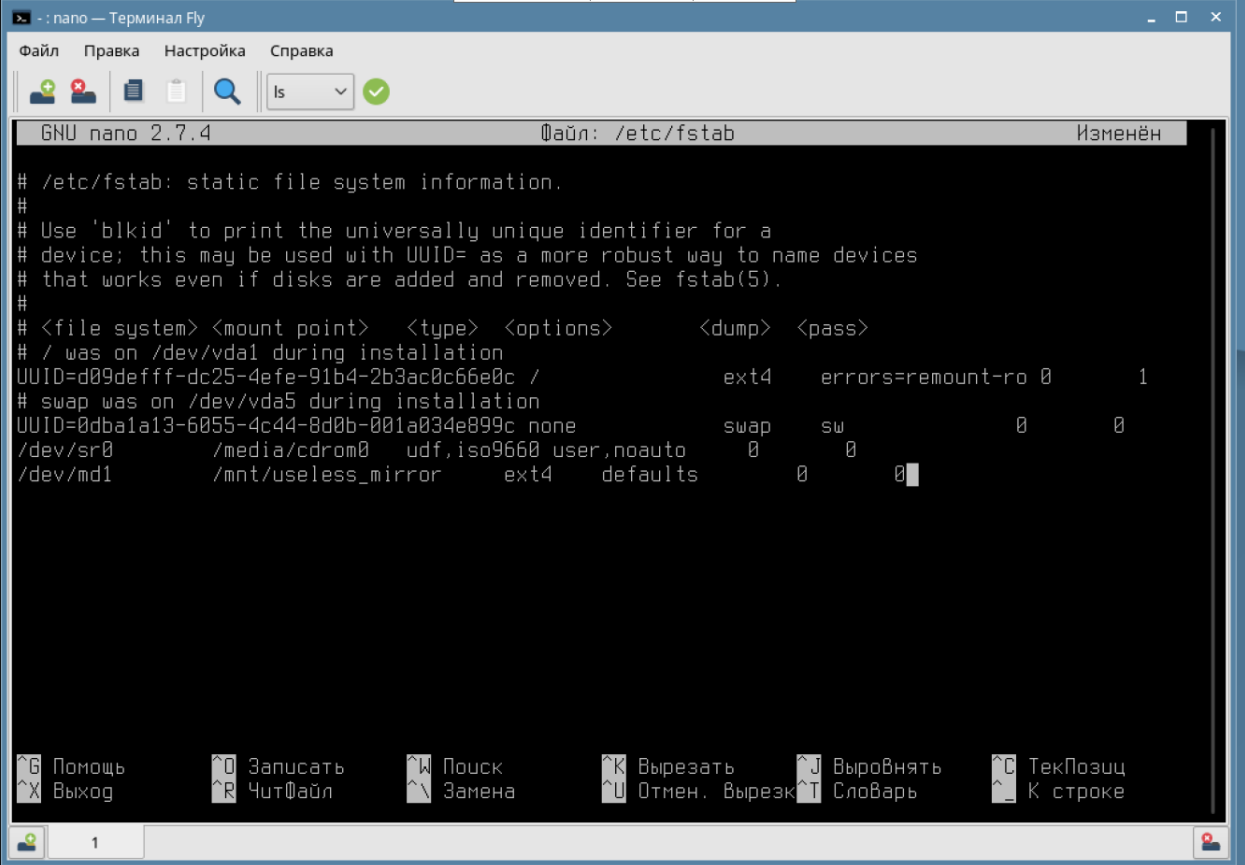
ВНИМАНИЕ!

Ошибки в данном файле приведут к невозможности корректной загрузки системы.

Добавьте в конец файла строчку

```
/dev/md127 /mnt/family_mirror ext4 defaults,nofail 0 0
```

Вместо `family` в `family_mirror` подставьте **свою фамилию** транслитом строчными (маленькими) буквами.



```
GNU nano 2.7.4 Файл: /etc/fstab Изменён
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/vda1 during installation
UUID=d09defff-dc25-4efe-91b4-2b3ac0c66e0c / ext4 errors=remount-ro 0 1
# swap was on /dev/vda5 during installation
UUID=0dba1a13-6055-4c44-8d0b-001a034e899c none swap sw 0 0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
/dev/md1 /mnt/useless_mirror ext4 defaults 0 0
```

Сохраните файл и завершите редактирование.

Теперь необходимо создать саму папку — точку монтирования.

```
mkdir /mnt/family_mirror
```

Для монтирования файловых систем, указанных в настройках системы, нет необходимости прописывать параметры целиком — достаточно указать лишь точку монтирования. Проверим корректность содержимого файла и возможность монтирования файловой системы командой

```
mount /mnt/family_mirror
```

И здесь, и выше вместо family в family_mirror подставьте **свою фамилию** транслитом строчными (маленькими) буквами.

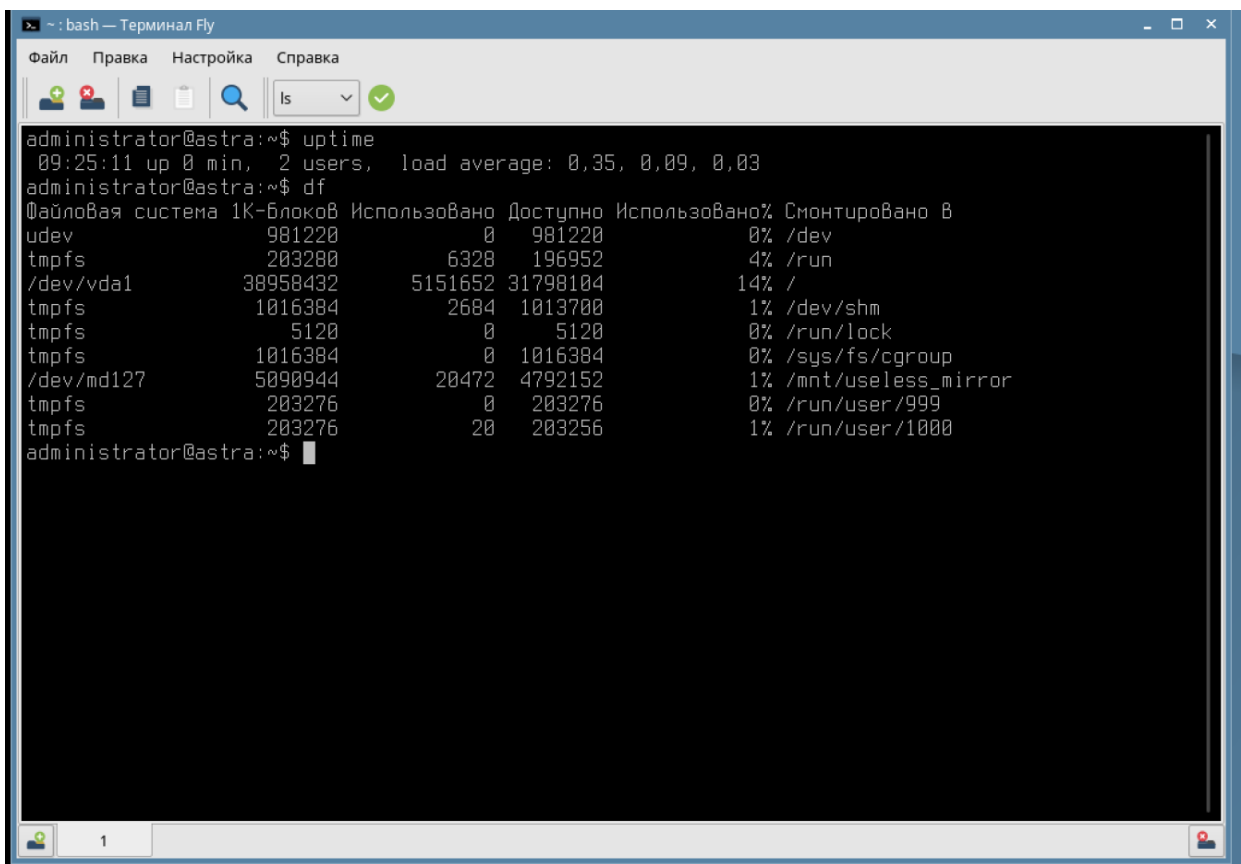
В случае отсутствия ошибок, команда mount ничего не выводит, поэтому проверим результаты командами mount и df

```
--: bash — Терминал Fly
Файл  Правка  Настройка  Справка
ls
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/lib/systemd/systemd-cgroups-agent,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=31,prgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=16259)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
mqueue on /dev/mqueue type mqueue (rw,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
fusectl on /sys/fs/fuse/connections type fusectl (rw,relatime)
configfs on /sys/kernel/config type configfs (rw,relatime)
tmpfs on /run/user/999 type tmpfs (rw,nosuid,nodev,relatime,size=203276k,mode=700,uid=999,gid=999,inode64)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=203276k,mode=700,uid=1000,gid=1000,inode64)
/dev/md1 on /mnt/useless_mirror type ext4 (rw,relatime)
root@astra: /home/administrator#
```

```
--: bash — Терминал Fly
Файл  Правка  Настройка  Справка
ls
root@astra: /home/administrator# df
Файловая система 1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
udev              982024          0    982024          0% /dev
tmpfs             203280         6320    196960          4% /run
/dev/vda1         38958432      5150088    31799668        14% /
tmpfs            1016384        2896    1013488          1% /dev/shm
tmpfs             5120           0         5120           0% /run/lock
tmpfs            1016384          0    1016384          0% /sys/fs/cgroup
tmpfs            203276          0    203276          0% /run/user/999
tmpfs            203276          20    203256          1% /run/user/1000
/dev/md1          5090944        20472    4792152          1% /mnt/useless_mirror
root@astra: /home/administrator#
```

И там, и там в списке должны присутствовать устройство /dev/md127 и соответствующая точка монтирования /mnt/family_mirror

Перезагрузим систему, чтобы убедиться, что все работает корректно. командами `uptime` и `df` подтвердите корректность работы



```
administrator@astra:~$ uptime
09:25:11 up 0 min,  2 users,  load average: 0,35, 0,09, 0,03
administrator@astra:~$ df
Файловая система 1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
udev                981220      0    981220         0% /dev
tmpfs               203280      6328    196952         4% /run
/dev/vda1          38958432   5151652  31798104        14% /
tmpfs              1016384    2684   1013700         1% /dev/shm
tmpfs               5120       0     5120           0% /run/lock
tmpfs              1016384    0   1016384         0% /sys/fs/cgroup
/dev/md127         5090944    20472   4792152         1% /mnt/useless_mirror
tmpfs              203276     0    203276         0% /run/user/999
tmpfs              203276     20   203256         1% /run/user/1000
administrator@astra:~$
```

Теперь перейдём к компоненту LVM2.

Он позволяет работать с множеством дисков, создавать тонкие, толстые, зеркальные, кэширующие и пр. тома. Поэтому в его случае имеем достаточно много абстракций и подготовительной работы.

Для начала, разрешим ему использовать оставшийся диск — `sdc`

```
pvcreate /dev/sdc
```

Теперь необходимо создать группу томов. Для нее необходимо указать название и включенные в неё физические тома.

```
vgcreate storage /dev/sdc
```

Далее создадим набор тонких томов из 5 штук. Для создания тонких томов необходим пул тонких томов, состоящий из толстого тома пространства и толстого тома метаданных. Создайте толстый том

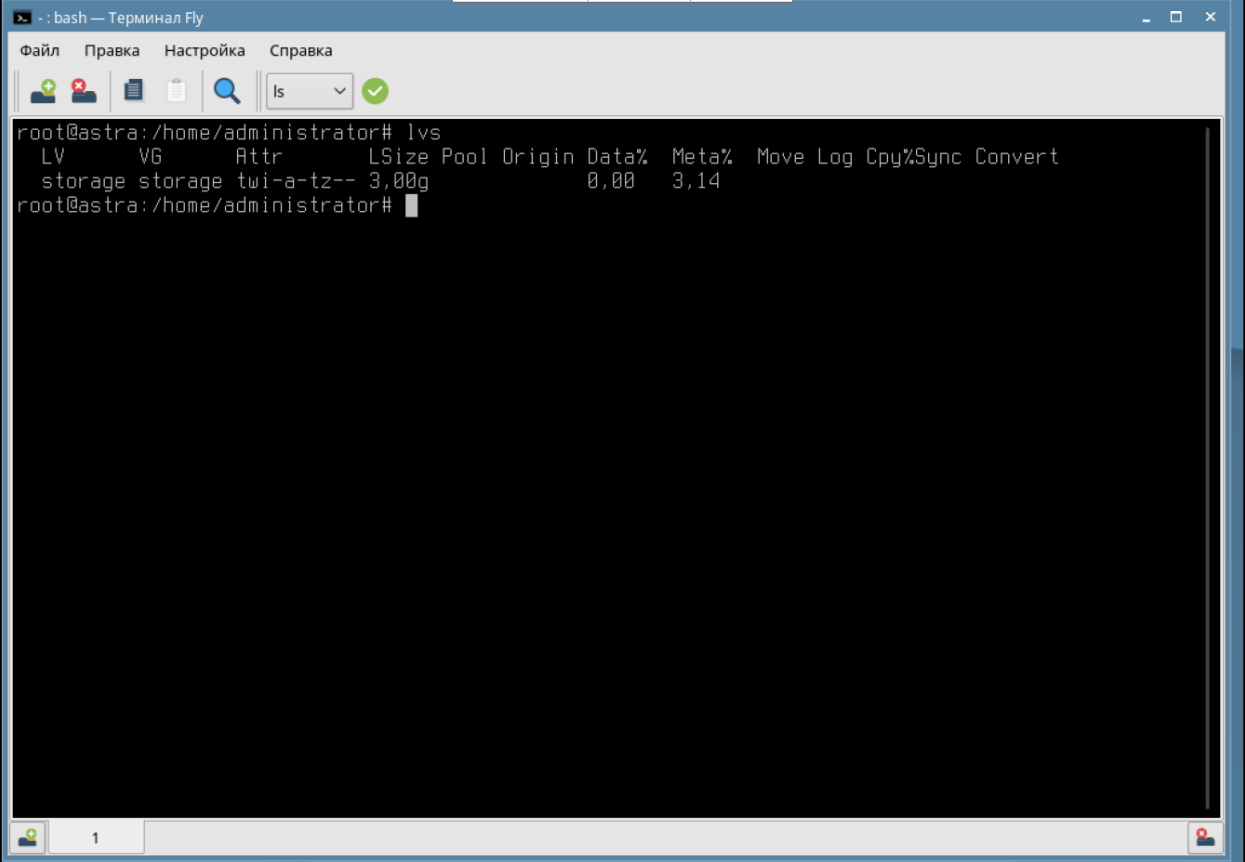
пространства объёмом 4 ГиБ и толстый том метаданных объёмом 0.5 ГиБ.

Соответствующие команды подсмотрите в справочной документации

```
man lvmthin
```

ПОДСКАЗКА:

Проверьте результат создания тома командой `lvs`.



```
root@astra:/home/administrator# lvs
LV      VG      Attr      LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
storage storage twi-a-tz-- 3,00g          0,00  3,14
root@astra:/home/administrator#
```

Теперь создадим 5 тонких томов с именами vol1 vol2 vol3 vol4 vol5. Размеры должны быть 1ГиБ, 2ГиБ, 3ГиБ, 4ГиБ и 5ГиБ.

```
lvcreate -V 1G -n vol1 --thinpool storage/storage
```

Остальные команды составьте самостоятельно.

Проверьте результат создания командой `lvs`

```
root@astra:/home/administrator# lvs
LV      VG      Attr      LSize Pool   Origin Data%  Meta%  Move Log Cpy%Sync Convert
storage storage twi---tz-- 3,00g storage 0,00  3,14
vol1    storage Vwi-a-tz-- 1,00g storage 0,00
vol2    storage Vwi-a-tz-- 2,00g storage 0,00
vol3    storage Vwi-a-tz-- 3,00g storage 0,00
vol4    storage Vwi-a-tz-- 4,00g storage 0,00
vol5    storage Vwi-a-tz-- 5,00g storage 0,00
root@astra:/home/administrator#
```

Теперь необходимо создать файловые системы и настроить автоматическое монтирование томов.

(Дальнейший пример использует только один том, вам по аналогии необходимо настроить все пять).

Создаём файловую систему

```
mkfs.ext4 /dev/storage/vol1
```

Создаём точку подключения

```
mkdir /mnt/vol1
```

Добавляем в /etc/fstab строчку с файловой системой

```
/dev/storage/vol1 /mnt/vol1 ext4 defaults,nofail 0 0
```

Примонтируем vol1

```
mount /mnt/vol1
```

Перезагружаемся, показываем корректность конфигурации командами `uptime`, `lvs` и `df`


```
administrator@astra:~$ sudo su
root@astra:/home/administrator# uptime
09:51:54 up 1 min, 2 users, load average: 0,21, 0,09, 0,03
root@astra:/home/administrator# lvs
LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert
storage storage twi-aotz-- 3,00g
vol1 storage Vwi-aotz-- 1,00g storage 4,78
vol2 storage Vwi-aotz-- 2,00g storage 4,76
vol3 storage Vwi-aotz-- 3,00g storage 3,72
vol4 storage Vwi-aotz-- 4,00g storage 3,19
vol5 storage Vwi-aotz-- 5,00g storage 2,88
root@astra:/home/administrator# df
Файловая система 1K-блоков Использовано Доступно Использовано% Смонтировано в
udev 981220 0 981220 0% /dev
tmpfs 203280 6432 196848 4% /run
/dev/vda1 38958432 5153428 31796328 14% /
tmpfs 1016384 2684 1013700 1% /dev/shm
tmpfs 5120 0 5120 0% /run/lock
tmpfs 1016384 0 1016384 0% /sys/fs/cgroup
/dev/md127 5090944 20472 4792152 1% /mnt/useless_mirror
/dev/mapper/storage-vol4 4062912 16376 3820440 1% /mnt/vol4
/dev/mapper/storage-vol2 1998672 6144 1871288 1% /mnt/vol2
/dev/mapper/storage-vol1 999320 2564 927944 1% /mnt/vol1
/dev/mapper/storage-vol3 3030800 9216 2847916 1% /mnt/vol3
/dev/mapper/storage-vol5 5095040 20472 4796040 1% /mnt/vol5
tmpfs 203276 0 203276 0% /run/user/999
tmpfs 203276 20 203256 1% /run/user/1000
root@astra:/home/administrator#
```

Заполните файл отчета «Шаблон для практической 10». Прикрепите его в СДО с названием «ПР10_Фамилия_Группа», где в названии будет указана ваша фамилия и группа.

Данный отчет должен содержать скриншоты выполнения работы (замените скриншотом слово <..скриншот..> в соответствующем пункте).

На **ВСЕХ** скриншотах, которые вы делаете, должно быть видно ваше ФИО и группу (для этого откройте блокнот и запишите их там), текущую дату и время и номер ВМ.

Не забудьте выключить виртуальную машину после себя (Пуск – Завершение работы).

Ответьте на теоретические вопросы:

1) Что делает команда `mkfs.ext4` ? Какие еще варианты могут быть, кроме `.ext4`? Что они означают?