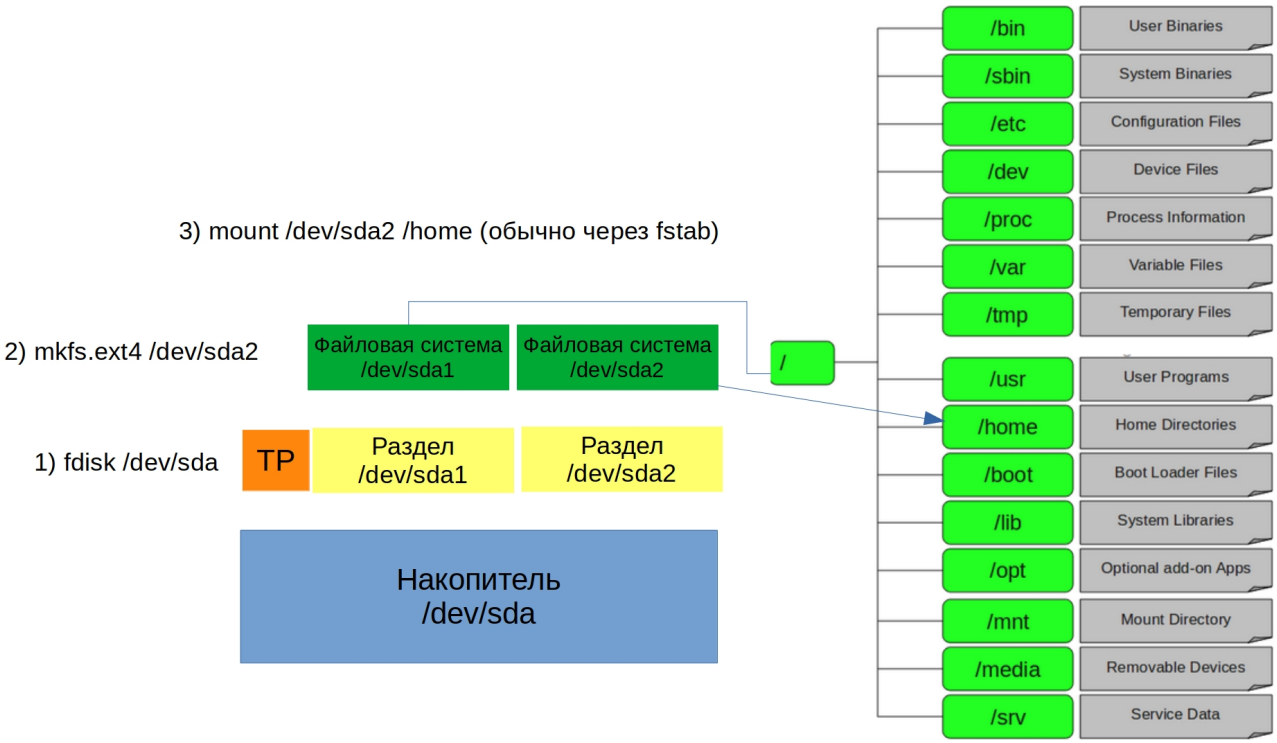
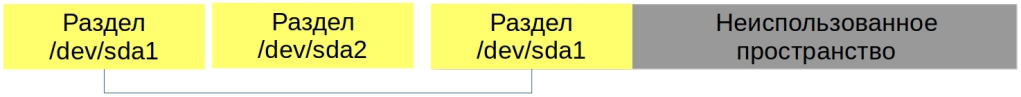


Стандартную схему работы с дисками и файловыми системами мы разобрали – создаём раздел с помощью fdisk, создаём на разделе файловую систему с помощью mkfs и монтируем файловую систему куда нужно.

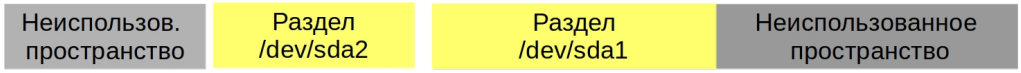
Но представьте себе ситуацию, когда у вас на каком-то разделе перестало хватать места. Если этот раздел крайний и после него есть свободное место – то его можно увеличить. А что если раздел не крайний, если после него есть ещё?

Нужно увеличить sda1

1) Переносим все блоки sda1 в свободное пространство



2) Увеличиваем sda1

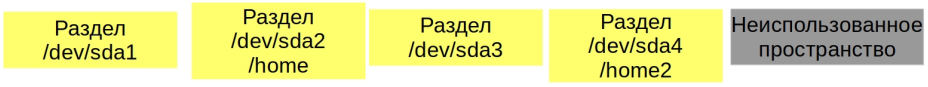


Занимает много времени и требует остановки работы

Стандартные таблицы разделов в этом плане ограничены, для раздела указывается только первый и последний сектор, то есть нельзя указать, чтобы раздел начался в одном месте, в другом закончился, а потом в третьем продолжился.

Нужно больше места для файлов пользователей

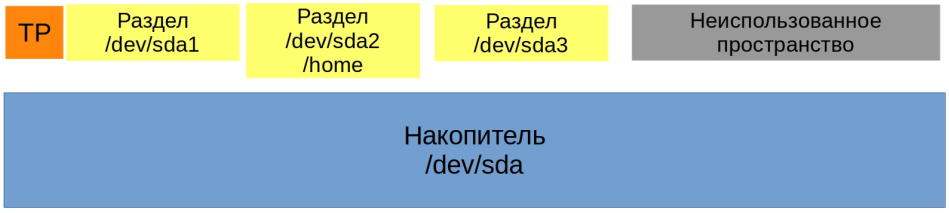
1) Создаём sda4, создаём файловую систему и монтируем в /home2



2) Переносим домашние директории некоторых пользователей в /home2

usermod user10 -m -d /home2/user10

Придётся держать часть файлов там, часть файлов тут, неудобно и некрасиво

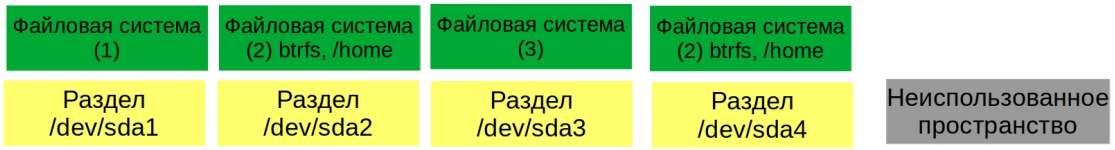


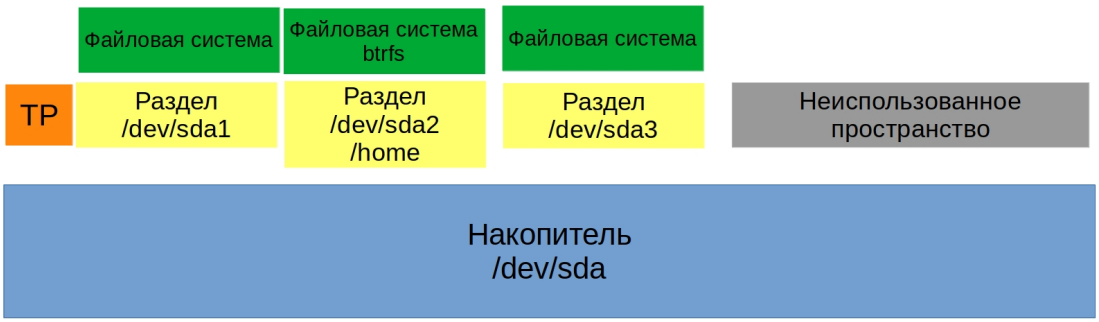
В некоторых случаях это не проблема – можно создать ещё один раздел, на нём поднять новую файловую систему и примонтировать куда-то. Допустим, создать директорию /home2, примонтировать туда новую файловую систему и перенести домашние директории некоторых пользователей в эту директорию. Просто нельзя несколько файловых систем монтировать в одну директорию. Точнее можно, но обходными, ненадёжными путями, чего стоит избегать.

Но иногда встречается софт, которому нужно хранить все свои файлы в одной директории, допустим, базу данных. И вот настаёт день, когда на разделе, где лежат данные, перестаёт хватать места. Примонтировать новую файловую систему в эту директорию не получится. Можно купить диск большего объёма и перенести на него файлы. Но, если там терабайты данных, перенос займёт много времени, а сервис при этом будет недоступен. И потом получится, что на первом диске место освободилось, но мы его не можем использовать. А завтра второй диск забьётся, опять купим новый диск большего объёма, опять перенесём на него файлы, и теперь у нас и первый и второй диск не будут использоваться.

Нужно больше места для файлов пользователей

1) Создаём sda4, с помощью btrfs расширяем файловую систему





Но мы монтируем файловую систему, так? И, если таблицы разделов не позволяют, можно ли на уровне файловой системы сделать так, чтобы одна файловая система могла работать на нескольких разделах и дисках? Да, действительно, есть такие файловые системы – допустим, btrfs. Но это относительно молодые файловые системы, пока не используемые повсеместно. Однако, вполне может быть, что через пару лет все файловые системы будут поддерживать возможность работы на нескольких разделах и дисках.

Хорошо, есть ли что-нибудь проверенное годами, что-то выше таблицы разделов, но ниже файловой системы? Да, есть такое и оно называется LVM – менеджер логических томов. Он позволяет вам объединять несколько разделов и дисков чтобы создать общее пространство. На основе этого общего пространства вы можете создавать логические разделы, на которых вы создадите файловую систему, примонтируете и всё как обычно.

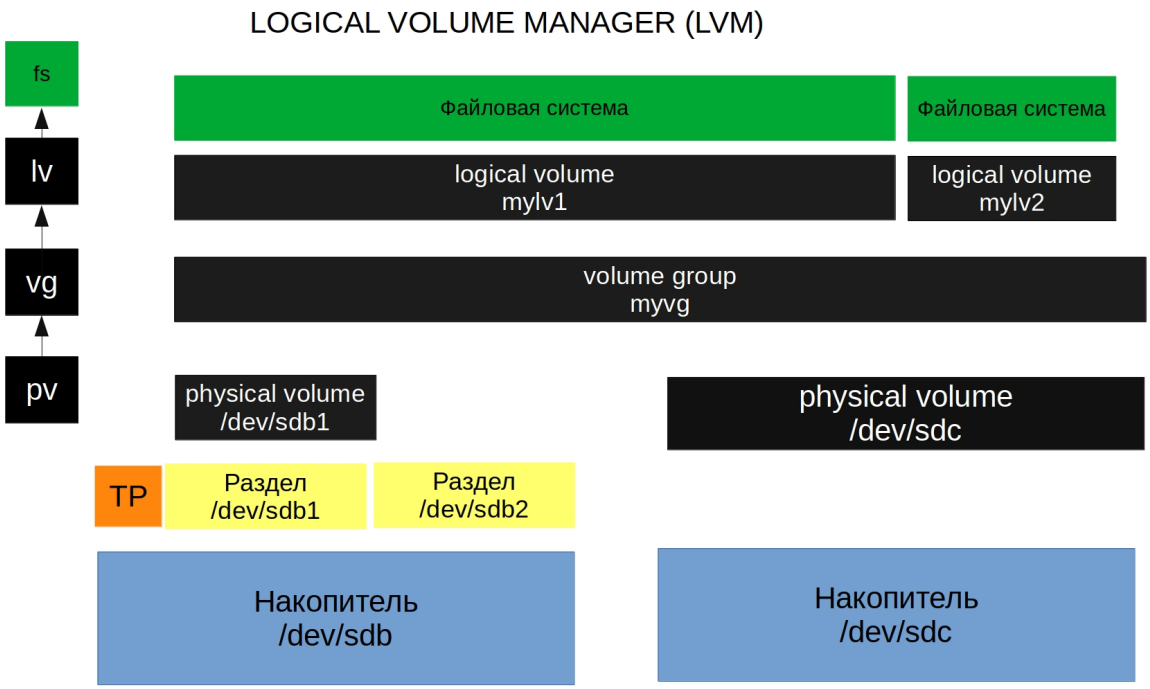


Схема довольно простая – ваши разделы и диски становятся физическими томами – physical volume – pv. Общее пространство, которое формирует группа физических томов называется группой томов – volume group – vg. А разделы, которые создаются внутри этой группы – логическими томами – logical volume – lv. Физический том – группа томов – логический том. Несложно, просто нужно пару раз попрактиковаться. При этом LVM позволяет добавлять разделы и диски в группу томов и увеличивать логические тома с файловыми системами без необходимости останавливать работу. Логические тома ещё можно уменьшать при необходимости, но для этого надо уменьшить файловую систему, а это не всегда просто. Это далеко не весь функционал LVM, но давайте пока немного попрактикуемся.

```
[user@centos8 ~]$ df -h
Filesystem                   Size  Used Avail Use% Mounted on
devtmpfs                     886M     0  886M   0% /dev
tmpfs                        914M     0  914M   0% /dev/shm
tmpfs                        914M  9.6M  904M   2% /run
tmpfs                        914M     0  914M   0% /sys/fs/cgroup
/dev/mapper/cl_centos8-root   17G  5.5G   12G  32% /
/dev/sda1                    976M  272M  638M  30% /boot
tmpfs                        183M  1.2M  182M   1% /run/user/42
tmpfs                        183M  9.1M  174M   5% /run/user/1000
/dev/sr0                      59M   59M     0 100% /run/media/user/VBox_GAs_6.1.
```

Для начала, я восстановил снапшот виртуалки, чтобы /home не был на диске sdc, так как нам понадобятся 2 диска для тестов - df -h. Вы можете восстановить снапшот, перенести /home обратно, либо добавить 2 новых диска.

```
[user@centos8 ~]$ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.32.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): g
Created a new GPT disklabel (GUID: 0E76E16E-C210-714F-AE26-EB6ECE51D37A).

Command (m for help): n
Partition number (1-128, default 1):
First sector (2048-2097118, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-2097118, default 2097118): +200M

Created a new partition 1 of type 'Linux filesystem' and of size 200 MiB.

Command (m for help): n
Partition number (2-128, default 2):
First sector (411648-2097118, default 411648):
Last sector, +sectors or +size{K,M,G,T,P} (411648-2097118, default 2097118): +400M

Created a new partition 2 of type 'Linux filesystem' and of size 400 MiB.
```

```
Command (m for help): p
Disk /dev/sdb: 1 GiB, 1073741824 bytes, 2097152 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 0E76E16E-C210-714F-AE26-EB6ECE51D37A

Device      Start     End Sectors  Size Type
/dev/sdb1    2048  411647  409600  200M Linux filesystem
/dev/sdb2  411648 1230847  819200  400M Linux filesystem

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

Начнём с простого – создадим таблицу разделов на диске sdb и два раздела на 200 и 400 мегабайт - sudo fdisk /dev/sdb; g - n, enter, enter, +200M; n, enter, enter, +400M; p, w.

```
[user@centos8 ~]$ sudo pvcreate /dev/sdb1
  Physical volume "/dev/sdb1" successfully created.
[user@centos8 ~]$ sudo pvs
  PV         VG         Fmt  Attr PSize   PFree
  /dev/sda2  cl_centos8 lvm2 a--  <19.00g      0
  /dev/sdb1             lvm2 ---  200.00m 200.00m
[user@centos8 ~]$ sudo pvdisplay /dev/sdb1
  "/dev/sdb1" is a new physical volume of "200.00 MiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdb1
  VG Name
  PV Size               200.00 MiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               UhB0l1-5CK9-By3e-z2bq-jyKa-7H46-I1QIWh
```

Хотя следующий шаг можно пропустить, я этого делать не буду, чтобы было понятнее. Вспоминаем последовательность – physical volume, volume group и logical volume. Все три нам нужно создать по порядку. И так, physical volume – это pv, создать – create - pvcreate и указываем раздел или диск, который будет использоваться – sudo pvcreate /dev/sdb1. Смотрим вывод – success – значит всё окей. У нас теперь есть физический том, посмотреть его мы можем с помощью sudo pvs; sudo pvdisplay /dev/sdb1. Как видите, тут помимо sdb1 есть ещё sda2 – это потому что при установке система сама создала LVM, но мы его трогать не будем.

```
[user@centos8 ~]$ sudo vgcreate myvg /dev/sdb1
  Volume group "myvg" successfully created
[user@centos8 ~]$ sudo vgs
  VG         #PV #LV #SN Attr   VSize   VFree
  cl_centos8   1   2   0 wz--n- <19.00g      0
  myvg         1   0   0 wz--n- 196.00m 196.00m
```

Physical volume у нас есть, дальше нужно создать группу томов – volume group – vgcreate. Для этого нужно дать ей имя и указать physical volume, один или несколько – sudo vgcreate myvg /dev/sdb1; sudo vgs. Опять видим success – всё окей.

```
[user@centos8 ~]$ sudo vgdisplay myvg
  --- Volume group ---
  VG Name               myvg
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               196.00 MiB
  PE Size               4.00 MiB
  Total PE              49
  Alloc PE / Size       0 / 0
  Free  PE / Size       49 / 196.00 MiB
  VG UUID               xqsUh5-kMmb-OX9Q-541S-8i12-kgA4-ezsLAM
```

Как посмотреть? Правильно - sudo vgdisplay myvg. Опять же, у нас тут есть volume group, который создался при установке – cl_centos8 и наша группа – myvg. Отсюда же можем посмотреть информацию о ней, допустим, сколько свободного места – почти 200 мегабайт. Тут ещё есть PE – physical extents. Это блоки, которыми оперирует LVM. Как видите, они у нас по 4 мебибайта и их у нас в сумме 49.

```
[user@centos8 ~]$ sudo lvcreate myvg -n mylv -L 200M
  Volume group "myvg" has insufficient free space (49 extents): 50 required.
[user@centos8 ~]$ sudo lvcreate myvg -n mylv -l 80%FREE
  Logical volume "mylv" created.
[user@centos8 ~]$ sudo lvs
  LV   VG         Attr       LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  root cl_centos8 -wi-ao---- <17.00g
  swap cl_centos8 -wi-ao----   2.00g
  mylv myvg       -wi-a----- 156.00m
```

Последний шаг – logical volume – lvcreate. Тут у нас опций чуть больше – нужно указать в каком volume group мы хотим создать наш логический том, указать имя для logical volume и указать размер – sudo lvcreate myvg -n (name – имя) mylv -L (размер) 200M - sudo lvcreate myvg -n mylv -L 200M. Как видите, у нас не получается выдать 200, потому что 1 блок ушёл на метаданные для LVM. Мы можем указать 196 или поступить по другому – выдать свободное место по процентам. Допустим, выдать всё свободное место, либо 80 процентов свободного - для этого ключ -l - sudo lvcreate myvg -n mylv -l 80%FREE; sudo lvs.

```
EXAMPLES
       Create a striped LV with 3 stripes, a stripe size  of  8KiB  and  a  size  of
       100MiB.  The LV name is chosen by lvcreate.
       lvcreate -i 3 -I 8 -L 100m vg00

       Create a raid1 LV with two images, and a useable size of 500 MiB. This opera-
       tion  requires  two  devices,  one  for  each  mirror  image.  RAID  metadata
       (superblock and bitmap) is also included on the two devices.
       lvcreate --type raid1 -m1 -L 500m -n mylv vg00

       Create  a  mirror  LV  with  two images, and a useable size of 500 MiB.  This
       operation requires three devices: two for mirror images and one  for  a  disk
       log.
       lvcreate --type mirror -m1 -L 500m -n mylv vg00
```

Кстати, важное замечание – если открыть man по любой из предыдущих команд lvm, допустим, man lvcreate и спуститься в самый низ, там есть примеры использования. И если вы вдруг забудете какой-то ключ, то эти примеры помогут вам вспомнить.

```
[user@centos8 ~]$ sudo lvdisplay myvg/mylv
  --- Logical volume ---
  LV Path                /dev/myvg/mylv
  LV Name                mylv
  VG Name                myvg
  LV UUID                riMLeI-28bT-PYNf-gnup-ze9L-gxbl-StUF0v
  LV Write Access        read/write
  LV Creation host, time centos8, 2021-03-15 14:02:16 +0400
  LV Status              available
  # open                 0
  LV Size                156.00 MiB
  Current LE             39
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:2
```

А сейчас, давайте посмотрим, что у нас получилось – sudo lvdisplay myvg/mylv. И так, тут у нас появился путь - /dev/myvg/mylv.

```
[user@centos8 ~]$ sudo mkfs.ext4 /dev/myvg/mylv
[sudo] password for user:
mke2fs 1.45.4 (23-Sep-2019)
Creating filesystem with 159744 1k blocks and 40000 inodes
Filesystem UUID: de9adc0c-b8f7-42dd-934d-23a6f504ccef
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

Именно сюда мы поставим файловую систему с помощью mkfs - sudo mkfs.ext4 /dev/myvg/mylv.

```
[user@centos8 ~]$ ll /dev/myvg/mylv
lrwxrwxrwx. 1 root root 7 Mar 15 14:18 /dev/myvg/mylv -> ../dm-2
[user@centos8 ~]$ ll /dev/dm-*
brw-rw----. 1 root disk 253, 0 Mar 15 13:57 /dev/dm-0
brw-rw----. 1 root disk 253, 1 Mar 15 13:57 /dev/dm-1
brw-rw----. 1 root disk 253, 2 Mar 15 14:18 /dev/dm-2
[user@centos8 ~]$ sudo lsblk -f /dev/myvg/mylv
NAME      FSTYPE LABEL UUID                                   MOUNTPOINT
myvg-mylv ext4         de9adc0c-b8f7-42dd-934d-23a6f504ccef
```

Сам же этот файл - ll /dev/myvg/mylv - является символической ссылкой на /dev/dm-2 - ll /dev/dm-*, который, как и sda, генерируется при обнаружении устройств. А значит, если говорить про fstab, /dev/dm-2 там указывать нельзя. Можно указать UUID - sudo lsblk -f /dev/myvg/mylv - или /dev/myvg/mylv.

```
[user@centos8 ~]$ tail -3 /etc/fstab
/dev/mapper/cl_centos8-root /                       xfs     defaults        0 0
UUID=c607eb50-9842-4b18-ba8e-2d1139960594 /boot                   ext4    defaults        1 2
/dev/mapper/cl_centos8-swap swap                    swap    defaults        0 0
[user@centos8 ~]$ ll /dev/mapper/
total 0
lrwxrwxrwx. 1 root root       7 Mar 15 13:57 cl_centos8-root -> ../dm-0
lrwxrwxrwx. 1 root root       7 Mar 15 13:57 cl_centos8-swap -> ../dm-1
crw-------. 1 root root 10, 236 Mar 15 13:57 control
lrwxrwxrwx. 1 root root       7 Mar 15 14:18 myvg-mylv -> ../dm-2
```

Но, при определённых условиях, о которых вы можете почитать по ссылке, оба варианта могут создавать проблемы. Поэтому лучший способ – способ, который предлагает сам RedHat – уже указан в fstab — tail -3 /etc/fstab - /dev/mapper. Если посмотреть - ll /dev/mapper - можно увидеть, что, как и /dev/myvg/mylv, тут символические ссылки.

```
/dev/mapper/cl_centos8-root /                       xfs     defaults
UUID=c607eb50-9842-4b18-ba8e-2d1139960594 /boot                   ext4
/dev/mapper/cl_centos8-swap swap                    swap    defaults

/dev/mapper/myvg-mylv  /mydata  ext4 noatime 0 0
```

В целях совместимости рекомендую использовать вариант именно с /dev/mapper, потому что на старых системах с другими вариантами могут быть проблемы - sudo nano /etc/fstab, /dev/mapper/myvg-mylv /mydata ext4 noatime 0 0.

```
[user@centos8 ~]$ sudo mkdir /mydata
[user@centos8 ~]$ sudo mount /mydata
[user@centos8 ~]$ df -h /mydata
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/myvg-mylv  148M  1.6M  135M   2% /mydata
[user@centos8 ~]$
```

Примонтируем и убедимся, что всё работает - sudo mkdir /mydata; sudo mount /mydata, df -h /mydata.

```
[user@centos8 ~]$ sudo pvcreate /dev/sdc
  Device /dev/sdc excluded by a filter.
[user@centos8 ~]$ sudo wipefs -a /dev/sdc
/dev/sdc: 8 bytes were erased at offset 0x00000200 (gpt): 45 46 49 20 50 41 52 54
/dev/sdc: 8 bytes were erased at offset 0x3ffffe00 (gpt): 45 46 49 20 50 41 52 54
/dev/sdc: 2 bytes were erased at offset 0x000001fe (PMBR): 55 aa
/dev/sdc: calling ioctl to re-read partition table: Success
[user@centos8 ~]$ sudo pvcreate /dev/sdc
  Physical volume "/dev/sdc" successfully created.
[user@centos8 ~]$
```

Предположим, у нас перестало хватать места и мы решили добавить ещё один диск – sdc. Так, у нас есть volume group, состоящий из одного physical volume – раздела sdb1. Я хочу полностью отдать весь диск sdc volume группе myvg. Первый шаг – создаю physical volume - sudo pvcreate /dev/sdc. Но вместо привычного success я тут вижу excluded by a filter. LVM заботливо отказывается использовать весь диск, так как там уже есть таблица разделов. Чтобы избавиться от неё я использую утилиту wipefs с ключом -a - sudo wipefs -a /dev/sdc. И теперь всё у меня создаётся как надо - sudo pvcreate /dev/sdc.

```
[user@centos8 ~]$ sudo vgextend myvg /dev/sdc
  Volume group "myvg" successfully extended
[user@centos8 ~]$ sudo vgdisplay myvg
  --- Volume group ---
  VG Name               myvg
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  3
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               1
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               <1.19 GiB
  PE Size               4.00 MiB
  Total PE              304
  Alloc PE / Size       39 / 156.00 MiB
  Free  PE / Size       265 / <1.04 GiB
  VG UUID               st1kGh-BY4o-NbM1-aNWC-pRZB-0Lse-Vc1M0k
```

Дальше мне нужно расширить volume группу и добавить в неё новый диск. Расширить это extend, а значит пишем - sudo vgextend myvg /dev/sdc. Видим success и смотрим sudo vgdisplay myvg. Теперь в myvg 2 physical volume, 304 блока и гигабайт свободного места. Но это пока что группа, нужно ещё увеличить logical volume и файловую систему.

```
[user@centos8 ~]$ df -h /mydata
Filesystem             Size  Used Avail Use% Mounted on
/dev/mapper/myvg-mylv  148M  1.6M  135M   2% /mydata
[user@centos8 ~]$ sudo lvextend myvg/mylv -L +500M
[sudo] password for user:
  Size of logical volume myvg/mylv changed from 156.00 MiB (39 extents) to 656.00 MiB (16
  Logical volume myvg/mylv successfully resized.
[user@centos8 ~]$ sudo lvs
  LV   VG         Attr       LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  root cl_centos8 -wi-ao---- <17.00g
  swap cl_centos8 -wi-ao----   2.00g
  mylv myvg       -wi-ao---- 656.00m
```

Да, если просто увеличить раздел или логический том – файловая система об изменениях не узнает. Поэтому нужно проделать 2 операции, но можно и одной командой. И так, смотрим сколько у нас свободного места на файловой системе - df -h /mydata – 100 с лишним мегабайт. Теперь пишем - sudo lvextend myvg/mylv -L +500M. Видим success – значит всё получилось. Смотрим размер logical volume – sudo lvs – 600 с лишним мегабайт.

```
[user@centos8 ~]$ df -h /mydata/
Filesystem             Size  Used Avail Use% Mounted on
/dev/mapper/myvg-mylv  148M  1.6M  135M   2% /mydata
[user@centos8 ~]$ sudo resize2fs /dev/myvg/mylv
[sudo] password for user:
resize2fs 1.45.4 (23-Sep-2019)
Filesystem at /dev/myvg/mylv is mounted on /mydata; on-line resizing required
old_desc_blocks = 2, new_desc_blocks = 6
The filesystem on /dev/myvg/mylv is now 671744 (1k) blocks long.

[user@centos8 ~]$ df -h /mydata/
Filesystem             Size  Used Avail Use% Mounted on
/dev/mapper/myvg-mylv  632M  2.5M  599M   1% /mydata
```

Смотрим df -h /mydata – 100 с лишним, ничего не изменилось. Файловая система изменений не увидела. Но об этом ей можно сказать – sudo resize2fs /dev/myvg/mylv. Теперь опять проверяем – df -h /mydata – всё как надо.

```
[user@centos8 ~]$ df -h /mydata/
Filesystem             Size  Used Avail Use% Mounted on
/dev/mapper/myvg-mylv  632M  2.5M  599M   1% /mydata
[user@centos8 ~]$ sudo lvextend myvg/mylv -L +100M -r
  Size of logical volume myvg/mylv changed from 656.00 MiB (164 extents) to 756.00 MiB
.
  Logical volume myvg/mylv successfully resized.
resize2fs 1.45.4 (23-Sep-2019)
Filesystem at /dev/mapper/myvg-mylv is mounted on /mydata; on-line resizing required
old_desc_blocks = 6, new_desc_blocks = 6
The filesystem on /dev/mapper/myvg-mylv is now 774144 (1k) blocks long.

[user@centos8 ~]$ df -h /mydata/
Filesystem             Size  Used Avail Use% Mounted on
/dev/mapper/myvg-mylv  729M  2.5M  692M   1% /mydata
```

Но можно сразу увеличить размер файловой системы при увеличении logical volume - тот же lvextend но с ключом -r - sudo lvextend myvg/mylv -L +100M -r; df -h /mydata. Как видите, никаких лишних действий.

Таким образом, мы смогли увеличить нашу файловую систему и в дальнейшем мы всегда сможем продолжать так делать, без необходимости останавливать работу, просто добавляя новые разделы или диски в нашу volume группу. При этом нам не важно, где заканчивается один lv и начинается другой – нет таких ограничений, как со стандартными разделами. Что касается удаления, то всё также просто – lvremove, vgremove и pvremove. В момент удаления логические тома не должны быть никуда примонтированы. Есть ещё много разных базовых команд с LVM – уменьшить размер, удалить physical volume из volume группы, переместить данные с одного physical volume на другой и много всего – но их много, они ситуативны, и всё разбирать в азах бессмысленно.

Но как я говорил, это далеко не весь функционал LVM-а. Как вы заметили, я стараюсь не выделять всё свободное пространство на logical volume. С одной стороны потому, что когда понадобится, увеличить размер файловой системы получится без труда. А вот если мне вдруг перестанет хватать места и я захочу уменьшить один из разделов – это будет сложнее. С другой стороны – свободное место нужно для снапшотов.

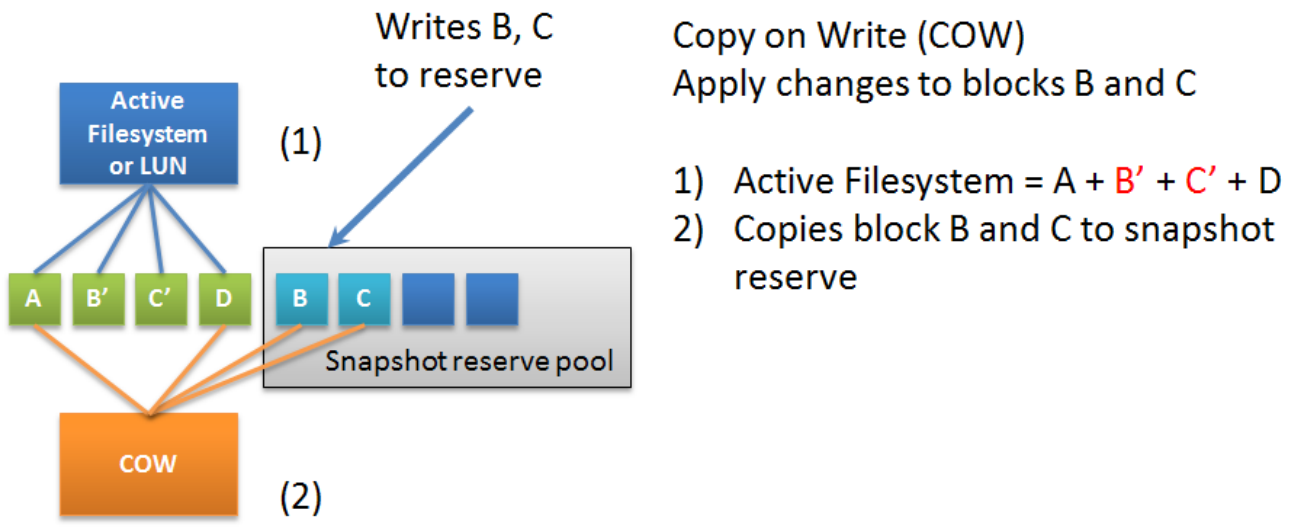
Вы уже знаете, что такое снапшот – снимок текущего состояния. Мы делали снапшоты для виртуалки - сняли снапшот после установки системы, и, если вдруг мы где-то что-то испортим, всегда можем вернуться к прежнему состоянию. Это как сохранение в играх. В виртуальной инфраструктуре гипервизоры позволяют нам делать снапшоты. Когда нет гипервизора, допустим, это система, установленная на физический сервер или ваш домашний компьютер – снапшот не сделаешь, но можно делать бэкапы. Но, в случае проблем, восстановление из бэкапа занимает много времени – нужно перенести данные из одного накопителя в другой, это сильно зависит от скорости накопителя. В целом, восстановление из бэкапа чуть ли не последний вариант, к которому прибегают администраторы.

Да и даже восстановление снапшота в виртуальной инфраструктуре не всегда приятно – для этого нужно выключить виртуалку – а если на ней какой-нибудь тяжелый софт, который долго запускается, или, скажем, виртуалку не трогали годами – есть шанс, что всплывёт куча других проблем. Тут нам на помощь приходит тот же самый LVM, который сам может делать снапшоты. Тут всё немного проще – вам не нужно перезагружать виртуалку и можно делать на физическом сервере. Всё что нужно чтобы восстановить – отмонтировать файловую систему. Да,

если у вас проблемы в корневой файловой системе, что-то там после обновления отвалилось, без перезагрузки не обойтись. Если проблемы с данными какого-нибудь сервиса вам придётся остановить его, чтобы отмонтировать файловую систему с данными и восстановить снапшот. Но это в любом случае быстрее, легче и нередко можно обойтись без перезагрузки.

```
[user@centos8 ~]$ sudo nano /mydata/file
[sudo] password for user:
[user@centos8 ~]$ cat /mydata/file
Before snapshot
[user@centos8 ~]$ sudo lvcreate -s -n mysnapshot -L 100M myvg/mylv
  Logical volume "mysnapshot" created.
[user@centos8 ~]$ sudo lvs
  LV         VG         Attr       LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  root       cl_centos8 -wi-ao---- <17.00g
  swap       cl_centos8 -wi-ao----   2.00g
  mylv       myvg       owi-aos--- 756.00m
  mysnapshot myvg       swi-a-s--- 100.00m      mylv   0.01
```

И так, прежде чем сделаем снапшот, создадим файл в /mydata, на который будем ориентироваться - sudo nano /mydata/file; Before snapshot; cat /mydata/file. Теперь создадим снапшот, нам для этого понадобится та же команда lvcreate с опцией -s - снапшот. Пишем sudo lvcreate -s -n (имя снапшота) mysnapshot -L 100M (размер снапшота) myvg/mylv — sudo lvcreate -s -n mysnapshot -L 100M myvg/mylv; sudo lvs. Снапшот создан.



У вас, возможно, возникли два вопроса – что за размер снапшота и почему так мгновенно? LVM использует технологию copy-on-write – копирование при изменении. Это работает так – мы создаём какое-то место для снапшота на диске – допустим на 100 мегабайт. Когда мы что-то меняем на исходном logical volume – старые блоки перед изменением копируются в это специальное место, в итоге на оригинальном lv новые блоки, а в выделенной области старые версии этих блоков. В случае, если у нас прям все блоки заменятся на исходном lv, размер снапшота должен быть размером с исходный логический том, ну, чуть больше, для всяких метаданных. А в большинстве случаев меняются не все блоки, а только часть, поэтому размер снапшота может быть меньше, т.е. нужно думать, а как много измениться на исходной lv, пока я буду держать снапшот.

```
[user@centos8 ~]$ sudo lvs myvg/mysnapshot
  LV         VG   Attr       LSize   Pool Origin Data%  Meta%
  mysnapshot myvg swi-a-s--- 100.00m      mylv   0.01
[user@centos8 ~]$ sudo nano /mydata/file
[user@centos8 ~]$ cat /mydata/file
After snapshot
[user@centos8 ~]$ sudo lvs myvg/mysnapshot
  LV         VG   Attr       LSize   Pool Origin Data%  Meta%
  mysnapshot myvg swi-a-s--- 100.00m      mylv   0.03
```

Теперь изменим наш файл — sudo lvs myvg/mysnaphost; sudo nano /mydata/file; After snapshot; cat /mydata/file. Смотрим lvs - sudo lvs myvg/mysnaphost - видим, что у снапшота используемое пространство немного изменилось.

```
[user@centos8 ~]$ sudo mount /dev/myvg/mysnapshot /mnt
[user@centos8 ~]$ cat /mnt/file
Before snapshot
[user@centos8 ~]$ cat /mydata/file
After snapshot
[user@centos8 ~]$
```

Кстати, мы можем примонтировать его и увидеть наши файлы в их исходном виде - sudo mount /dev/myvg/mysnapshot /mnt, cat /mnt/file, cat /mydata/file.

```
[user@centos8 ~]$ sudo umount /mnt /mydata
[user@centos8 ~]$ sudo lvconvert --merge myvg/mysnapshot
  Merging of volume myvg/mysnapshot started.
  myvg/mylv: Merged: 100.00%
[user@centos8 ~]$ sudo mount /mydata
[user@centos8 ~]$ sudo cat /mydata/file
Before snapshot
[user@centos8 ~]$
```

Снапшот можно удалить как и обычный логический том – sudo lvremove myvg/mysnapshot, но я этого делать не буду. Вместо этого, давайте восстановимся из этого снапшота. Для этого нам понадобится отмонтировать снапшот и исходный том - sudo umount /mnt /mydata. Далее соединить снапшот с исходным lv – sudo lvconvert –merge myvg/mysnapshot.

Дальше снапшот удалится и мы сможем примонтировать наш lv как раньше и проверить файл - sudo mount /mydata, cat /mydata/file, sudo lvs.

В целом снапшоты увеличивают количество записей на диске, поэтому на активно используемых томах они будут снижать производительность диска, особенно когда снапшотов несколько. Поэтому их стоит делать перед какими-то изменениями – допустим, перед обновлением системы. А потом, если всё прошло успешно, их удалять. Также снапшоты помогают с бэкапом – во время снапшота сохраняется текущее состояние файловой системы, из-за чего софту для бэкапа легче работать с этим самым снапшотом, а не исходной файловой системой, на которую постоянно может идти запись.

Если же в целом говорить про LVM, это отличный инструмент, который даёт администратору возможность более гибко работать с дисками. В то же время, это дополнительный слой абстракции и проблемы с LVM решать сложнее, чем проблемы с обычными разделами. Однако, больше практики, самостоятельного изучения и всяких экспериментов помогут вам лучше освоить этот инструмент. Больше информации про LVM вы можете найти по ссылке. Но, в рамках основ, вам достаточно знать базовые операции – как создать physical volume, volume group, logical volume, как расширить vg и lv с учётом файловой системы, ну и как работать со снапшотами.