

Лабораторная работа №3 : Работа с LVM

1. Посмотрите список существующих Групп Томов (VG) LVM. Задокументируйте: Сколько свободных и занятых физических экстендов (PE) есть в этой группе. Какой размер физического экстенда?

```
# vgsdisplay
```

```
--- Volume group ---
VG Name                cl
System ID
Format                 lvm2
Metadata Areas         1
Metadata Sequence No   3
VG Access               read/write
VG Status               resizable
MAX LV                 0
Cur LV                 2
Open LV                 2
Max PV                 0
Cur PV                 1
Act PV                 1
VG Size                 7.00 GiB
PE Size                 4.00 MiB
Total PE                1791
Alloc PE / Size         1791 / 7.00 GiB
Free PE / Size          0 / 0
VG UUID                 Wc8BwU-rJeP-24KF-TU4D-LeMb-oQyU-H7DcsX
```

2. Какие физические тома (PV) входят в эту группу? Посмотрите и задокументируйте их свойства.

```
# pvdisplay
```

```
--- Physical volume ---
PV Name                /dev/sda2
VG Name                cl
PV Size                 7.00 GiB / not usable 3.00 MiB
Allocatable             yes (but full)
PE Size                 4.00 MiB
Total PE                1791
Free PE                 0
Allocated PE            1791
PV UUID                 ykkdNU-XdqW-LvqU-ospT-BC1D-T3fE-oLyhHC
```

3. Какие логические тома (LV) находятся в этой группе? Посмотрите и задокументируйте их свойства.

```
# lvdisplay cl
```

```
--- Logical volume ---
LV Path                /dev/cl/swap
LV Name                swap
VG Name                cl
LV UUID                YL0SBP-n7pK-EqJS-pC1M-d9Cu-1t2E-rTRRca
LV Write Access         read/write
LV Creation host, time localhost.localdomain, 2017-03-17 23:15:06 +0300
LV Status                available
# open                  2
LV Size                 820.00 MiB
Current LE              205
Segments                1
Allocation              inherit
Read ahead sectors      auto
- currently set to      256
Block device            253:1

--- Logical volume ---
LV Path                /dev/cl/root
LV Name                root
VG Name                cl
LV UUID                MFH9p0-5Kaj-slcW-JJF0-V121-tsRp-McooJ5
LV Write Access         read/write
LV Creation host, time localhost.localdomain, 2017-03-17 23:15:07 +0300
LV Status                available
# open                  1
LV Size                 6.20 GiB
Current LE              1586
Segments                1
Allocation              inherit
Read ahead sectors      auto
- currently set to      256
Block device            253:0
```

4. Сконфигурируйте предназначенные для LVM разделы Ваших трех дисков как физические тома LVM. (pvcreate)

```
# pvcreate /dev/vda7
# pvcreate /dev/vdb7
# pvcreate /dev/vdc7
```

```
[root@localhost ~]# pvcreate /dev/vda7
Physical volume "/dev/vda7" successfully created.
[root@localhost ~]# pvcreate /dev/vdb7
Physical volume "/dev/vdb7" successfully created.
[root@localhost ~]# pvcreate /dev/vdc7
Physical volume "/dev/vdc7" successfully created.
```

5. Так же сконфигурируйте для включения в LVM RAID1 созданный во второй лабораторной работе (/dev/md1)

```
# pvcreate /dev/md1
```

```
[root@localhost ~]# pvcreate /dev/vda7
Physical volume "/dev/vda7" successfully created.
[root@localhost ~]# pvcreate /dev/vdb7
Physical volume "/dev/vdb7" successfully created.
[root@localhost ~]# pvcreate /dev/vdc7
Physical volume "/dev/vdc7" successfully created.
[root@localhost ~]# pvcreate /dev/md1
WARNING: xfs signature detected on /dev/md1 at offset 0. Wipe it? [y/n]: y
Wiping xfs signature on /dev/md1.
Physical volume "/dev/md1" successfully created.
[root@localhost ~]#
```

6. Создайте группу томов vg01, используя физический том на RAID1

```
# vgcreate vg01 /dev/md1
```

```
[root@localhost ~]# vgcreate vg01 /dev/md1
Volume group "vg01" successfully created
```

7. Создайте группу томов vg02, используя два физических тома на основе разделов первого и второго дисков.

```
# vgcreate vg02 /dev/vda7 /dev/vdb7
```

```
[root@localhost ~]# vgcreate vg02 /dev/vda7 /dev/vdb7
Volume group "vg02" successfully created
```

8. Используйте команды `vgdisplay` и `pvdisplay` для проверки статуса Вашей новой группы томов и физического тома. Сколько физических томов сейчас в Вашей группе томов? Сколько логических томов сейчас в Вашей группе томов? Какой размер экстенда?

```
# vgdisplay vg01
```

В группе томов vg01 – один физический и ноль логических томов

```
# vgdisplay vg02
```

В группе томов vg02 – два физических и ноль логических томов

```

[root@localhost ~]# vgdisplay vg01
--- Volume group ---
VG Name                vg01
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 PU                0
Cur PU               1
Act PU                1
VG Size               248.00 MiB
PE Size               4.00 MiB
Total PE              62
Alloc PE / Size       0 / 0
Free PE / Size        62 / 248.00 MiB
VG UUID               fGNiUy-NZP2-1k4K-AQvo-mdMZ-W1k2-CGP1bU

[root@localhost ~]# vgdisplay vg02
--- Volume group ---
VG Name                vg02
System ID
Format                 lvm2
Metadata Areas         2
Metadata Sequence No   1
VG Access              read/write
VG Status              resizable
MAX LV                0
Cur LV               0
Open LV               0
Max PU                0
Cur PU               2
Act PU                2
VG Size               496.00 MiB
PE Size               4.00 MiB
Total PE              124
Alloc PE / Size       0 / 0
Free PE / Size        124 / 496.00 MiB
VG UUID               Uerrja-Yxaw-n8wx-BwfT-x9ps-C0Ff-QD0vzC

```

9. В группе vg01 попробуйте создать том линейный том data_vol1 и том с чередованием strp_vol1 размером по 9 МБ каждый. Объясните результат. Какие файлы устройств будут соответствовать созданным томам? Какую информацию о томах можно по ним получить? (В том числе используя команду dmsetup)

```
# lvcreate -L 9M -n data_vol1 vg01
```

```

[root@localhost ~]# lvcreate -L 9M -n data_vol1 vg01
Rounding up size to full physical extent 12.00 MiB
Logical volume "data_vol1" created.

```

```
# lvcreate -i1 -l64 -L 9M -n strp_vol1 vg01
```

```

[root@localhost ~]# lvcreate -i1 -l64 -L 9M -n strp_vol1 vg01
Ignoring stripesize argument with single stripe.
Rounding up size to full physical extent 12.00 MiB
Logical volume "strp_vol1" created.

```

Поставить -i2 и больше нельзя, так как количество разбиений не может превышать количество физических томов в группе, а у нас в vg01 только один физ. том - /dev/md1. Поэтому был проигнорирован аргумент размера чередования (64K) и был создан обычный **линейный** логический том.

Информация по созданному логическому тому:

```
# dmsetup info
```

```

Name:                vg01-strt_vol1
State:               ACTIVE
Read Ahead:         256
Tables present:     LIVE
Open count:         0
Event number:       0
Major, minor:       253, 3
Number of targets:  1
UUID: LUM-fGNIUyNZP21k4KAQvomdMZW1k2CGP1bUumCe4hfCCtBG0MmYdY4KKZ4oUxigNHgS

Name:                vg01-data_vol1
State:               ACTIVE
Read Ahead:         256
Tables present:     LIVE
Open count:         0
Event number:       0
Major, minor:       253, 2
Number of targets:  1
UUID: LUM-fGNIUyNZP21k4KAQvomdMZW1k2CGP1bU6XPEQERbfsXLdf3oD1JQ8DhxY1B1LGdI

```

10. Создайте логические тома ora_db и mysql_db размером по 9 МБ в группе vg02. Какого размера тома Вы получили и почему?

```

# lvcreate -L 9M -n ora_db vg02
# lvcreate -L 9M -n mysql_db vg02

```

```

[root@localhost ~]# lvcreate -L 9M -n ora_db vg02
Rounding up size to full physical extent 12.00 MiB
Logical volume "ora_db" created.
[root@localhost ~]# lvcreate -L 9M -n mysql_db vg02
Rounding up size to full physical extent 12.00 MiB
Logical volume "mysql_db" created.

```

Мы получили тома размером по 12МБ каждый, так как 9 мегабайт не делятся без остатка на размер экстенда(4Мб). Поэтому размер каждого тома был округлен до ближайшего числа, которое делится – 12 МБ. Получилось по 3 экстенда на том.

11. Создайте логический том размером 9 МБ с зеркалом в группе vg02. (Имя тома mir)

```

# lvcreate -L 9M -m1 -n mir vg02

```

```

[root@localhost ~]# lvcreate -L 9M -m1 -n mir vg02
Rounding up size to full physical extent 12.00 MiB
Logical volume "mir" created.

```

12. Какие том vg02/mir отображен на физические тома и физические экстенды? (используйте ключ -m команды lvsdisplay) Какого размера том Вы получили и почему? Какие три способа размещения журнала тома с зеркалированием есть в LVM? Какой был использован?

```

# lvsdisplay -m /dev/vg02/mir

```

```
[root@localhost ~]# lvdisplay -m /dev/vg02/mir
--- Logical volume ---
LV Path                /dev/vg02/mir
LV Name                 mir
VG Name                vg02
LV UUID                46U307-PpcQ-o0Ij-UxRV-YGTt-jH90-cdgkqe
LV Write Access         read/write
LV Creation host, time localhost.localdomain, 2017-05-15 20:48:54 +0300
LV Status               available
# open                  0
LV Size                 12.00 MiB
Current LE              3
Mirrored volumes        2
Segments                1
Allocation              inherit
Read ahead sectors      auto
- currently set to     256
Block device            253:10

--- Segments ---
Logical extents 0 to 2:
  Type                  raid1
  Monitoring            monitored
  Raid Data LV 0
    Logical volume      mir_rimage_0
    Logical extents     0 to 2
  Raid Data LV 1
    Logical volume      mir_rimage_1
    Logical extents     0 to 2
  Raid Metadata LV 0    mir_rmeta_0
  Raid Metadata LV 1    mir_rmeta_1
```

Мы получили том размером 12 мб из-за вышеупомянутого округления по экстендам. Был создан логический том mir с одним зеркалом: mir_rimage_0 и mir_rimage_1. Каждое из зеркал занимает по 3 логических экстенда (с номерами от 0 до 2). Каждое из зеркал имеет метаданные, которые по умолчанию занимают 512KB: mir_rmeta_0 и mir_rmeta_1.

pvdisplay -m /dev/vda7

```
[root@localhost ~]# pvdisplay -m /dev/vda7
--- Physical volume ---
PV Name                 /dev/vda7
VG Name                vg02
PV Size                 250.00 MiB / not usable 2.00 MiB
Allocatable             yes
PE Size                 4.00 MiB
Total PE                62
Free PE                 52
Allocated PE            10
PV UUID                 nsBP5e-kaDD-iZYf-mErU-awHu-eIj1-Hv2oeE

--- Physical Segments ---
Physical extent 0 to 2:
  Logical volume        /dev/vg02/ora_db
  Logical extents       0 to 2
Physical extent 3 to 5:
  Logical volume        /dev/vg02/mysql_db
  Logical extents       0 to 2
Physical extent 6 to 6:
  Logical volume        /dev/vg02/mir_rmeta_0
  Logical extents       0 to 0
Physical extent 7 to 9:
  Logical volume        /dev/vg02/mir_rimage_0
  Logical extents       0 to 2
Physical extent 10 to 61:
  FREE
```

На первом физическом томе из группы vg02 - /dev/vda7 были размещены: метаданные первого зеркального отражения mir_rmeta_0 (в физическом экстенде 6) и само первое зеркало mir_rimage_0 (в физических экстендах 7-9).

```
# pvdisplay -m /dev/vdb7
```

```
[root@localhost ~]# pvdisplay -m /dev/vdb7
--- Physical volume ---
PU Name                /dev/vdb7
VG Name                vg02
PU Size                250.00 MiB / not usable 2.00 MiB
Allocatable            yes
PE Size                4.00 MiB
Total PE               62
Free PE                58
Allocated PE           4
PV UUID                DnCu5Y-NP5i-G9h9-kLgW-qy0v-S4J7-0117eU

--- Physical Segments ---
Physical extent 0 to 0:
  Logical volume        /dev/vg02/mir_rmeta_1
  Logical extents        0 to 0
Physical extent 1 to 3:
  Logical volume        /dev/vg02/mir_rimage_1
  Logical extents        0 to 2
Physical extent 4 to 61:
  FREE
```

На втором физическом томе из группы vg02 - /dev/vdb7 были размещены: метаданные второго зеркального отражения mir_rmeta_1 (в физическом экстенсте 0) и само второе зеркало mir_rimage_1 (в физических экстенстах 1-3).

Четыре способа размещения лога:

- На отдельном физическом томе (тогда для создания зеркального логич. тома с -m 1 понадобится 3 физ. тома: 2 для самих данных и один для лога)
- На том же физ. томе, что и сами зеркальный данные, при использовании опции --alloc anywhere. Сокращает необходимое для зеркалирования количество физ. томов на один, но ухудшает производительность.
- В оперативной памяти, при вводе опции --corelog или --mirrorlog core. Это создает необходимость синхронизации зеркального тома при каждой перезагрузке.
- Сам лог так же может быть отзеркален (создан в двух экземплярах) при помощи опции --mirrorlog mirrored

В нашем случае лог был размещен в оперативной памяти, так как у нас всего 2 физических тома в группе vg02 и на них разместились сами зеркала, а для лога не осталось физ. тома. Об этом свидетельствует следующий скрин (лога на нем не видеть).

```
# lvs -a vg02
```

```
[root@localhost ~]# lvs -a vg02
LV      VG   Attr      LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
mir     vg02 rwi-a-r--- 12.00m                                     100.00
[mir_rimage_0] vg02 iwi-aor--- 12.00m
[mir_rimage_1] vg02 iwi-aor--- 12.00m
[mir_rmeta_0]  vg02 ewi-aor--- 4.00m
[mir_rmeta_1]  vg02 ewi-aor--- 4.00m
mysql_db  vg02 -wi-a----- 12.00m
ora_db    vg02 -wi-a----- 12.00m
```

13. Создайте логический том размером 9 МБ с чередованием на два тома с шагом в 4 КБ в группе vg02. (Имя тома stri2) Какого размер тома Вы получили и почему?

```
# lvcreate -i2 -l4 -L 9M -n stri2 vg02
```

```
root@localhost ~]# lvcreate -i2 -l4 -L 9M -n stri2 vg02
Rounding up size to full physical extent 12.00 MiB
Rounding size 12.00 MiB (3 extents) up to stripe boundary size 16.00 MiB (4 extents).
Logical volume "stri2" created.
```

Мы получили том размером 16МБ (12 Мб из-за округления исходных 9 МБ до делимого на 4МБ (12МБ, 3 экстенда). А 16МБ из-за того, что чередование будет происходить по 4КБ на каждый из физ. томов по очереди, таким образом, когда мы превысим объем в 2 экстенда (данные на них чередовались между 1 и 2 экстендами по 4КБ), нам нужно будет чередовать данные между следующими двумя экстендами (3 и 4), однако при 12МБ у нас только 3 экстенда и чередование не возможно – поэтому добавляется еще один экстенд (4МБ), что бы суммарное кол-во было четным = 4 и мы смогли чередовать.

14. Какие том vg02/stri2 отображен на физические тома и физические экстенды? (используйте ключ -m команды lvdisplay)

```
# pvdisplay -m /dev/vda7
```

```
# pvdisplay -m /dev/vdb7
```

```
root@localhost ~]# pvdisplay -m /dev/vda7
--- Physical volume ---
PV Name                /dev/vda7
VG Name                vg02
PV Size                250.00 MiB / not usable 2.00 MiB
Allocatable            yes
PE Size                4.00 MiB
Total PE               62
Free PE                50
Allocated PE           12
PV UUID                nsBP5e-kaDD-iZYf-mErU-awHu-eIj1-Hv2ocE

--- Physical Segments ---
Physical extent 0 to 2:
  Logical volume        /dev/vg02/ora_db
  Logical extents        0 to 2
Physical extent 3 to 5:
  Logical volume        /dev/vg02/mysql_db
  Logical extents        0 to 2
Physical extent 6 to 6:
  Logical volume        /dev/vg02/mir_rmeta_0
  Logical extents        0 to 0
Physical extent 7 to 9:
  Logical volume        /dev/vg02/mir_rimage_0
  Logical extents        0 to 2
Physical extent 10 to 11:
  Logical volume        /dev/vg02/stri2
  Logical extents        0 to 3
Physical extent 12 to 61:
  FREE
```

```
root@localhost ~]# pvdisplay -m /dev/vdb7
--- Physical volume ---
PV Name                /dev/vdb7
VG Name                vg02
PV Size                250.00 MiB / not usable 2.00 MiB
Allocatable            yes
PE Size                4.00 MiB
Total PE               62
Free PE                56
Allocated PE           6
PV UUID                DnCu5Y-NP5i-69h9-kLgW-qy0v-S4J7-0117eU

--- Physical Segments ---
Physical extent 0 to 0:
  Logical volume        /dev/vg02/mir_rmeta_1
  Logical extents        0 to 0
Physical extent 1 to 3:
  Logical volume        /dev/vg02/mir_rimage_1
  Logical extents        0 to 2
Physical extent 4 to 5:
  Logical volume        /dev/vg02/stri2
  Logical extents        0 to 3
Physical extent 6 to 61:
  FREE
```

Как видно на скриншотах, stri2 занимает два физ. экстенда на томе /dev/vda7 (10-11) и два экстенда на томе /dev/vdb7 (4-5). Данные чередуются между физ томами /dev/vda7 и /dev/vdb7.

15. Включите в группу vg02 физические тома на основе раздела третьего диска.

```
# vgextend vg02 /dev/vdc7
```

```
root@localhost ~]# vgextend vg02 /dev/vdc7
Volume group "vg02" successfully extended
```

16. Создайте логический том размером 9 МБ с чередованием на три тома с нагом в 4 КБ в группе vg02. (Имя тома strip3) Какого размера тома Вы получили и почему?

```
# lvcreate -i3 -l4 -L 9M -n strip3 vg02
```

```
root@localhost ~]# lvcreate -i3 -l4 -L 9M -n strip3 vg02
Rounding up size to full physical extent 12.00 MiB
Logical volume "strip3" created.
```

Мы получили размер в 3 экстенда (12МБ) из-за округления от 9МБ. Дальнейшего расширения не произошло потому, что чередование будет происходить по трем физ. томам, а 12 делится на 3. Следовательно на каждом из 3 физ. томов vg02 будет выделено по одному физ. экстенду для данного логического тома strip3. И чередоваться будет по 4КБ между этими 3 экстендами.

17. Увеличьте размер тома ora_db на 11 МБ, а тома mysql_db на 20% свободного места в группе. Какого размера стали тома ora_db и mysql_db в конечном итоге.

```
# lvresize -L +11M /dev/vg02/ora_db - 24 мб
```

```
# lvextend -L +20%FREE /dev/vg02/mysql_db — 144мб
```

```
root@localhost ~]# lvresize -L +11M /dev/vg02/ora_db
Rounding size to boundary between physical extents: 12.00 MiB.
Size of logical volume vg02/ora_db changed from 12.00 MiB (3 extents) to 24.00 MiB (6 extents).
Logical volume vg02/ora_db successfully resized.
root@localhost ~]# lvextend -l +20%FREE /dev/vg02/mysql_db
Size of logical volume vg02/mysql_db changed from 12.00 MiB (3 extents) to 144.00 MiB (36 extents).
Logical volume vg02/mysql_db successfully resized.
```

Том ora_db расширился до 24МБ (6 экстендов) из-за округления с добавляемого размера с 11МБ до 12МБ (т.к. размер экстенда 4МБ). Ну и, если добавить к 12 МБ еще 12, получается 24МБ и 6 экстендов.

А том mysql_db просто расширился на 20% свободного места, до 144МБ (36 экстендов).

18. Создайте пул thinpool для тонких томов в группе vg02, используйте для этого 50% свободного места в группе.

```
# lvcreate -l 50%FREE -T /dev/vg02/thinpool
```

```
root@localhost ~]# lvcreate -l 50%FREE -T /dev/vg02/thinpool
Using default stripesize 64.00 KiB.
Logical volume "thinpool" created.
```

19. Создайте тонкий логический том, с виртуальным размером 1ТБ. Командами lvdisplay и lvs проанализируйте результат.

```
# lvcreate -V 1024G -T -n thin_lv /dev/vg02/thinpool
```

```
root@localhost ~]# lvcreate -V 1024G -T -n thin_lv /dev/vg02/thinpool
Using default stripesize 64.00 KiB.
WARNING: Sum of all thin volume sizes (1.00 TiB) exceeds the size of thin pool vg02/thinpool and the size of whole volume group (744.00 MiB)!
For thin pool auto extension activation/thin_pool_autoextend_threshold should be below 100.
Logical volume "thin_lv" created.
```

```
root@localhost ~]# lvs vg02
```

LV	VG	Attr	LSize	Pool	Origin	Data%	Meta%	Move	Log	Cpy%	Sync	Convert
mir	vg02	rwi-a-r---	12.00m								100.00	
mysql_db	vg02	-wi-a-----	144.00m									
ora_db	vg02	-wi-a-----	24.00m									
stri2	vg02	-wi-a-----	16.00m									
strip3	vg02	-wi-a-----	12.00m									
thin_lv	vg02	Uwi-a-tz--	1.00t	thinpool		0.00						
thinpool	vg02	twi-aotz--	248.00m			0.00	0.98					


```

root@localhost ~]# lvsdisplay /dev/vg02/thin_lv
--- Logical volume ---
LV Path                /dev/vg02/thin_lv
LV Name                 thin_lv
VG Name                 vg02
LV UUID                 UGldy-mor2-2rGG-h3q2-Pb1r-vnzw-QBD101
LV Write Access         read/write
LV Creation host, time localhost.localdomain, 2017-05-15 23:12:53 +0300
LV Pool name            thinkpool
LV Status                available
# open                  0
LV Size                 1.00 TiB
Mapped size             0.00%
Current LE               262144
Segments                1
Allocation               inherit
Read ahead sectors      auto
- currently set to      256
Block device            253:17

```

Из результата команд видно, что наш логический том thin_lv успешно создан с виртуальным размером 1TB.

20. На томе mysql_db создайте, смонтируйте и заполните данными файловую систему

```

# mkfs.xfs /dev/vg02/mysql_db
# mkdir /mnt/mysql_db ; mount /dev/vg02/mysql_db /mnt/mysql_db
# cp -a /etc /mnt/mysql_db

```

21. Создайте тонкий снимок snap1 тома mysql_db

```

# lvcreate -L 9M -s -n snap1 vg02/mysql_db

```

```

[root@localhost ~]# lvcreate -L 9M -s -n snap1 vg02/mysql_db
Using default stripesize 64.00 KiB.
Rounding up size to full physical extent 12.00 MiB
WARNING: Sum of all thin volume sizes (1.00 TiB) exceeds the size of thin pools and the size of whole volume group (744.00 MiB)
)
For thin pool auto extension activation/thin_pool_autoextend_threshold should be below 100.
Logical volume "snap1" created.

```

22. Смонтируйте файловую систему на снимке

```

# mkdir /mnt/snap1
# mount -o nouuid /dev/vg02/snap1 /mnt/snap1

```

23. Сравните содержимое каталогов /mnt/mysql_db и /mnt/snap1

```

[root@localhost ~]# ls /mnt/mysql_db/
etc
[root@localhost ~]# ls /mnt/snap1/
etc

```

Мы видим что содержимое наших томов одинаковое.

24. Скопируйте в /mnt/mysql_db файл /var/log/messages. Повторно сравните содержимое каталогов /mnt/mysql_db и /mnt/snap1

```

# cp /var/log/messages /mnt/mysql_db

```

```

[root@localhost ~]# cp /var/log/messages /mnt/mysql_db/
[root@localhost ~]# ls /mnt/mysql_db/
etc messages
[root@localhost ~]# ls /mnt/snap1/
etc

```

Мы видим, что на в оригинальном томе появился файл messages, а в снимке нет (т.к. снимки сохраняют состояние том в конкретный момент времени, и не меняются при дальнейших изменениях тома, а размер снимков увеличивается по мере изменения оригинального тома, т.к. снимки хранят различия между текущим состоянием оригинального тома и его состоянием в момент «съемки»).

25. Сбросьте дисковый кэш командой **sync** и посмотрите информацию о **snap1** командой **lvs**. Что изменилось?

```
[root@localhost ~]# lvs vg02
LV          VG      Attr      LSize   Pool            Origin      Data%   Meta%   Move Log   Cpy%Sync Convert
mir         vg02    rwi-a-r--- 12.00m
mysql_db    vg02    owi-aos--- 144.00m
ora_db      vg02    -wi-a----- 24.00m
snap1       vg02    swi-aos--- 12.00m          mysql_db 17.84
stri2       vg02    -wi-a----- 16.00m
strip3      vg02    -wi-a----- 12.00m
thin_lv     vg02    Uwi-a-tz--  1.00t thinkpool      0.00
thinkpool   vg02    twi-aotz-- 248.00m        0.00    0.98

[root@localhost ~]#
[root@localhost ~]# sync
[root@localhost ~]# lvs vg02
LV          VG      Attr      LSize   Pool            Origin      Data%   Meta%   Move Log   Cpy%Sync Convert
mir         vg02    rwi-a-r--- 12.00m
mysql_db    vg02    owi-aos--- 144.00m
ora_db      vg02    -wi-a----- 24.00m
snap1       vg02    swi-aos--- 12.00m          mysql_db 17.84
stri2       vg02    -wi-a----- 16.00m
strip3      vg02    -wi-a----- 12.00m
thin_lv     vg02    Uwi-a-tz--  1.00t thinkpool      0.00
thinkpool   vg02    twi-aotz-- 248.00m        0.00    0.98
```

sync записывает данные, буферизованные в памяти, на диск. Буферизируются, например, измененные супер-блоки, измененные inode'ы, операции чтения и записи также откладываются. Буферизация должна быть реализована в ядре. Программа **sync** всего лишь делает системный вызов **sync**.

Ядро хранит данные в памяти во избежание частых (обычно медленных) дисковых операций чтения и записи. Это повышает производительность, но если компьютер аварийно завершает работу, данные могут быть утеряны, либо может быть повреждена файловая система. **sync** гарантирует, что все, что хранилось в памяти, будет записано на диск

26. Отключите файловую систему **/mnt/snap1** (**umount /mnt/snap1**), и удалите **snap1** командой **lvremove**

```
# umount /mnt/snap1
```

```
# lvremove /dev/vg02/snap1
```

```
[root@localhost ~]# umount /mnt/snap1/
[root@localhost ~]# lvremove vg02/snap1
Do you really want to remove active logical volume vg02/snap1? [y/n]: y
Logical volume "snap1" successfully removed
[root@localhost ~]#
```

27. Посмотрите сколько свободных физических экстендов есть в группе томов **vg02**. Командой **df -h** посмотрите размер файловой системы подключенной к **/mnt/mysql_db**. Увеличьте том **mysql_db** еще на 50% свободного места.

```
# vgdisplay vg02
```

```
# df -h /dev/mapper/vg02-mysql_db
```

```

[root@localhost ~]# vgdisplay vg02
--- Volume group ---
VG Name                vg02
System ID
Format                 lvm2
Metadata Areas         3
Metadata Sequence No   23
VG Access               read/write
VG Status               resizable
MAX LV                 0
Cur LV                 7
Open LV                 1
Max PV                 0
Cur PV                 3
Act PV                 3
VG Size                 744.00 MiB
PE Size                 4.00 MiB
Total PE                186
Alloc PE / Size         121 / 484.00 MiB
Free PE / Size           65 / 260.00 MiB
VG UUID                Uerrja-Yxaw-n8wx-Bwft-x9ps-C0Ff-QD0vzC

[root@localhost ~]# df -h /dev/mapper/vg02-mysql_db
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/vg02-mysql_db 141M  39M  103M  28% /mnt/mysql_db

```

Размер ФС 144МБ, занято 39МБ

Увеличим размер:

```
# lvextend -l +50%FREE vg02/mysql_db
```

```

[root@localhost ~]# lvextend -l +50%FREE vg02/mysql_db
WARNING: Sum of all thin volume sizes (1.00 TiB) exceeds the size of thin pools and the size of whole volume group (744.00 MiB)
)
For thin pool auto extension activation/thin_pool_autoextend_threshold should be below 100.
Size of logical volume vg02/mysql_db changed from 144.00 MiB (36 extents) to 276.00 MiB (69 extents).
Logical volume vg02/mysql_db successfully resized.

```

28. Посмотрите текущий размер тома mysql_db. На экстенды каких физических томов он отображен? Увеличился ли размер файловой системы /mnt/mysql_db?

```
# lvdisplay -m /dev/vg02/mysql_db
```

Размер ФС не изменился

```

[root@localhost ~]# lvdisplay -m /dev/vg02/mysql_db
--- Logical volume ---
LV Path                /dev/vg02/mysql_db
LV Name                mysql_db
VG Name                vg02
LV UUID                re2ZKR-NZ10-WsSa-OKTT-bxHW-9u9C-UUtxAJ
LV Write Access         read/write
LV Creation host, time localhost.localdomain, 2017-05-15 20:47:21 +0300
LV Status               available
# open                  1
LV Size                 276.00 MiB
Current LE              69
Segments                4
Allocation              inherit
Read ahead sectors      auto
- currently set to      256
Block device            253:3

--- Segments ---
Logical extents 0 to 2:
  Type                linear
  Physical volume      /dev/vda7
  Physical extents     3 to 5

Logical extents 3 to 35:
  Type                linear
  Physical volume      /dev/vda7
  Physical extents     16 to 48

Logical extents 36 to 45:
  Type                linear
  Physical volume      /dev/vda7
  Physical extents     52 to 61

Logical extents 46 to 68:
  Type                linear
  Physical volume      /dev/vdb7
  Physical extents     7 to 29

[root@localhost ~]# df -h /dev/mapper/vg02-mysql_db
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/vg02-mysql_db 141M  39M  103M  28% /mnt/mysql_db

```

29. Выполните команду `xfs_growfs /dev/vg02/mysql_db` Проверьте размер файловой системы `/mnt/mysql_db`

```
[root@localhost ~]# df -h /dev/mapper/vg02-mysql_db
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/vg02-mysql_db 141M      39M    103M   28% /mnt/mysql_db
[root@localhost ~]# xfs_growfs /dev/vg02/mysql_db
meta-data=/dev/mapper/vg02-mysql_db isize=512    agcount=4, agsize=9216 blks
         =                       sectsz=512    attr=2, projid32bit=1
         =                       crc=1        finobt=0 spinodes=0
data     =                       bsize=4096   blocks=36864, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal              bsize=4096   blocks=855, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                  extsz=4096   blocks=0, rtextents=0
data blocks changed from 36864 to 78656
[root@localhost ~]# df -h /dev/mapper/vg02-mysql_db
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/vg02-mysql_db 273M      39M    235M   15% /mnt/mysql_db
```

Размер ФС расширился до 273МБ.

30. Удалите логический том `data_vol1` в группе `vg01`.

```
# lvremove vg01/data_vol1
```

```
[root@localhost ~]# lvremove vg01/data_vol1
Do you really want to remove active logical volume vg01/data_vol1? [y/n]: y
Logical volume "data_vol1" successfully removed
```

31. Удалите группу томов `vg01`.

```
# vgremove vg01
```

```
[root@localhost ~]# vgremove vg01
Do you really want to remove volume group "vg01" containing 1 logical volumes? [y/n]: y
Do you really want to remove active logical volume vg01/strp_vol1? [y/n]: y
Logical volume "strp_vol1" successfully removed
Volume group "vg01" successfully removed
```

32. Что изменилось в свойствах физического тома, который ранее был включен в группу `vg01`?

```
[root@localhost ~]# pvdiskdisplay -m /dev/md1
"/dev/md1" is a new physical volume of "249.94 MiB"
--- NEW Physical volume ---
PU Name                /dev/md1
UG Name
PU Size                249.94 MiB
Allocatable            NO
PE Size                0
Total PE               0
Free PE                0
Allocated PE           0
PU UUID                PwIozf-UxZE-t0E4-loMA-sw1I-luc3-qdULGj
```

Мы видим, что было убрано разбиение физ. тома на физ. экстенды, так как группа томов была удалена. Следовательно, отражение логич. томов на физ. экстенды было очищено.