

Дискове, дялове и swap диск.

Дял (partition) – логическа част от реален диск.

Таблица на дяловете (partition table) – таблица описваща разположението и големината на дяловете на диска – **MBR (master boot record)** и **GPT (GUID partition table)**.

- **MBR** – primary дялове (максимум четири);
- extended дял (максимум – един) - логически дял;
- максимален размер на диска е 2 TB.
  - **GPT** – до 128 дяла;
- могат да бъдат до 9.4 милиарда TB в размер.
- нямат extended и логически дялове.

### Главни локации във файловата система:

/ → Дъното на директорното дърво.  
/var → Променлива локация, log files и динамичното съдържание е тук, като уеб сайтове.  
/home → Потребителската папка където се съдържат личните файлове.  
/boot → Зареждаща папка, обикновено е на друг дял. Линукс ядрото и други поддържащи файлове се намират там.  
/opt → локация за опционален софтуер, обикновено ползван от 3ти лица.  
Ентерпрайз апликациите използват доста тази папка.

### Swap space:

- Временно място което се държи като RAM, когато процент от рама е пълен, ядрото ще премести не толкова ползваните данни върху swap.
- Swap partition.
- Swap file (подобен на page file в windows) - доста по бавен спрямо цял партишън
- Размер на Swap като се наглася винаги гледайте да е между 1.5 и 2.0 по размера на RAM паметта ви.
- След като RAM паметта е станала евтина, е по хубаво да не под 50% от RAM.

### Партишъни:

- **/dev/sda** - първият драйв закачен към компютъра.
- **/dev/sda1** е почти целият диск, но най често се дели на **/dev/sda2** и **/dev/sda3**.
- **колкото повече са партишъните толкова по голям е шанса да има маунт точки.**
- Например можем да имаме **/dev/sda1, /dev/sda2, /dev/sda3** където sda1 е отделен за root directory, sda2 за home, sda3 за boot.

- **mount** - може да се ползва за закачане на партишъни или да покаже всички 'закачки' закачени когато е без опции.

```
nick@xmachine:~$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,noexec,relatime,size=8105376k,nr_inodes=2026344,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,noexec,relatime,size=1626712k,mode=755)
/dev/nvme0n1p5 on / type ext4 (rw,relatime,errors=remount-ro)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup2 on /sys/fs/cgroup/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
none on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
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/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=28,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=24902)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
/dev/nvme0n1p1 on /boot/efi type vfat (rw,relatime,fmask=0077,dmask=0077,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,nosuid,nodev,noexec,relatime)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=1626708k,mode=700,uid=1000,gid=1000)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
```

- **lsblk** - показва колко блок устройства има в системата и техните имена.

```
nick@xmachine:~$ lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
nvme0n1      259:0    0 953,9G  0 disk
├─nvme0n1p1 259:1    0   512M  0 part /boot/efi
├─nvme0n1p2 259:2    0     1K  0 part
└─nvme0n1p5 259:3    0 953,4G  0 part /
```

- **fdisk -l /dev/diskname** - може да покаже лист със всички партишъни и информация за диска.

```
nick@xmachine:~$ sudo fdisk -l
Disk /dev/nvme0n1: 953,89 GiB, 1024209543168 bytes, 2000409264 sectors
Disk model: SAMSUNG MZVLW1T0HMLH-000H1
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: dos
Disk identifier: 0x5d0226c6

Device      Boot  Start      End  Sectors  Size Id Type
/dev/nvme0n1p1 *    2048    1050623   1048576   512M b W95 FAT32
/dev/nvme0n1p2        1052670 2000408575 1999355906 953,4G 5 Extended
/dev/nvme0n1p5        1052672 2000408575 1999355904 953,4G 83 Linux
```

- `swapon --summary` показва кратко пояснение за swap паметта и употребата и същата информация може да намерите и в `/proc/swaps`.

```
nick@xmachine:~$ swapon --summary
Filename                                Type      Size      Used      Priority
/swapfile                              file      25165820  0          0
```

LVM абстрактен слой между физическите дискове и файловата система - позволява създаването на групи от дискове или дялове, които могат да бъдат сглобени в единична (или неограничена) файлова система

- може да се ползва за всеки mount point освен boot, защото GRUB не може да чете LVM метаданни.

- Гъвкавост - позволява преоразмеряване на обемите.

- Snapshots - позволява за моментни копия на вашия логически обем LVM позволява лесна промяна на размера на файловите системи online, както и възможност една файлова система да се разпростира на няколко физически диска.

- **Physical Volume (PV)** – може да бъде цял диск, дял или RAID масив
- **Volume Group (VG)** – административна единица, която обединява physical и logical volumes.
- **VG са изградени от PVs** , които от своя страна са разделени на Physical Extents (PE).
- **Logical Volume (LV)** – еквивалент на дисковите дялове (partition) при структура без LVM. LVs са блокови устройства , които са изградени от PEs от една и съща VG.
- **Physical Extent (PE)** - части от данни, които изграждат physical volumes. Тези PEs имат фиксиран размер за една VG. По подразбиране PE е 4MB, но може да се променя по време на създаване на VG.
- **Logical Extent (LE)** - части от данни, които изграждат logical volumes. Имат същия размер като PEs.

Създаване на логически обеми:

1. Използваме “lsblk” за да видим свободните устройства:

```
[nick@localhost ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   20G  0 disk
├─sda1       8:1    0    1G  0 part /boot
├─sda2       8:2    0   19G  0 part
│ └─cl-root 253:0    0   17G  0 lvm  /
└─cl-swap    253:1    0    2G  0 lvm  [SWAP]
sdb          8:16   0    8G  0 disk
sdc          8:32   0    1G  0 disk
sdd          8:48   0    1G  0 disk
sde          8:64   0    1G  0 disk
sr0         11:0    1 1024M  0 rom
```

2. Използваме “pvcreate”: зареждаме (PV) физически обем за използване от LVM:

```
[nick@localhost ~]$ sudo pvcreate /dev/sdb
[sudo] password for nick:
Physical volume "/dev/sdb" successfully created.
```

3. Използваме “vgcreate” за да направим нашата обемна група:

```
[nick@localhost ~]$ sudo vgcreate nashtagroupa /dev/sdb
A volume group called nashtagroupa already exists.
```

4. Използваме “lvcreate” за да направим нашият логически обем.

Ползваме -L за да укажем размера на LV и нашето име с -n, по принцип такова име автоматично ще бъде приложено от системата.

```
[nick@localhost ~]$ sudo lvcreate -L 6500M -n logicheskiobem nashtagroupa
Logical volume "logicheskiobem" created.
```

5. Прилагаме команда с цел създаване на файлова система, без нея система това което ще се случи е, че логическият обем ще бъде не използваем. Командата е “mkfs” и малко опций:

```
[nick@localhost ~]$ sudo mkfs -t ext4 /dev/nashtagroupa/logicheskiobem
mke2fs 1.45.4 (23-Sep-2019)
Creating filesystem with 1664000 4k blocks and 416160 inodes
Filesystem UUID: bdf190af-c52d-43e2-b12a-8e47b9c08aee
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

6. Така вече сме създали нашият обем ( диск, дял ) това място може да бъде използвано от нас. Тоест това което трябва да се случи е монтаж:

```
[nick@localhost ~]$ sudo mount /dev/nashtagroupa/logicheskiobem ~/mountvolume/
mount: /home/nick/mountvolume: /dev/mapper/nashtagroupa-logicheskiobem already mounted on /home/nick/mountvolume.
```

Управление на VG и LV:

- Преглед:

- **pvdisplay:** (показва атрибутите на PV (физическият обем) , като размер, размер на физическа степен, пространство, използвано за дескриптора на VG площ и др.).

```
[nick@localhost ~]$ sudo pvdisplay
--- Physical volume ---
PV Name                /dev/sda2
VG Name                cl
PV Size                <19.00 GiB / not usable 3.00 MiB
Allocatable            yes (but full)
PE Size                4.00 MiB
Total PE               4863
Free PE                0
Allocated PE           4863
PV UUID                IgHQdr-uNgE-YfPg-I82U-m62C-1cJx-JNeVC3
```

- **vgdisplay:** (показва атрибутите на VG (обемната група) и свързаните PV (физически) и LV (логически) ).

```
[nick@localhost ~]$ sudo vgdisplay
--- 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                <19.00 GiB
PE Size                4.00 MiB
Total PE               4863
Alloc PE / Size         4863 / <19.00 GiB
Free PE / Size          0 / 0
VG UUID                ESm2Vj-yx9D-LeyM-CXeZ-a7ds-bwQ3-Utk1HU
```



- **lvdisplay:** (показва информация за логическият обем).

```
[nick@localhost ~]$ sudo lvdisplay
--- Logical volume ---
LV Path                /dev/cl/swap
LV Name                 swap
VG Name                 cl
LV UUID                 jC0vEv-R4Y4-jnra-i0yJ-8Jnr-eic9-xhTm2e
LV Write Access         read/write
LV Creation host, time localhost, 2020-10-26 10:29:27 -0400
LV Status                available
# open                  2
LV Size                 2.00 GiB
Current LE               512
Segments                1
Allocation               inherit
Read ahead sectors      auto
- currently set to      8192
Block device            253:1

--- Logical volume ---
LV Path                /dev/cl/root
LV Name                 root
VG Name                 cl
LV UUID                 Tij7i0-YZXd-clxD-zidW-mYj1-zuWW-dDzGBi
LV Write Access         read/write
LV Creation host, time localhost, 2020-10-26 10:29:27 -0400
LV Status                available
# open                  1
LV Size                 <17.00 GiB
Current LE               4351
Segments                1
Allocation               inherit
Read ahead sectors      auto
- currently set to      8192
Block device            253:0
```

- **lvmdiskscan:** (показва устройствата които могат да бъдат използвани като физически обеми).

```
[nick@localhost ~]$ sudo lvmdiskscan
/dev/sda1 [ 1.00 GiB]
/dev/sda2 [ <19.00 GiB] LVM physical volume
/dev/sdb [ 8.00 GiB]
/dev/sdc [ 1.00 GiB]
/dev/sdd [ 1.00 GiB]
/dev/sde [ 1.00 GiB]
4 disks
1 partition
0 LVM physical volume whole disks
1 LVM physical volume
```

- **pvs:** (показва информация за физическият обем).

```
[nick@localhost ~]$ sudo pvs
PV          VG Fmt  Attr PSize  PFree
/dev/sda2   cl  lvm2 a--  <19.00g  0
```

- **vgs:** (показва информация за групираният обем).

```
[nick@localhost ~]$ sudo vgs
VG #PV #LV #SN Attr   VSize  VFree
cl   1   2   0 wz--n- <19.00g  0
```

- **lvs:** (показва информация за логическите обеми).

```
[nick@localhost ~]$ sudo lvs
LV   VG Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
root cl -wi-ao---- <17.00g
swap cl -wi-ao---- 2.00g
```

- Оразмеряване:

**vgreduce:** Премахва физически обем(и) от обемна група (можем да ги наричаме и токове)

**vgextend:** Добавяш физически обем(и) към обемна група.

**pvresize:** Преоразмерява физическия обем.

**lvreduce:** Променя размера на логическия обем в случая намаля.

**lvextend:** Увеличава размера на логическия обем.

**lvresize:** Преоразмерява логическия обем.

**resize2fs:** Преоразмерява файлови системи от тип ext2, ext3 или ext4.

**xfs\_growfs:** Увеличава размера на XFS файлова система.

Добавяне на нов физически диск към съществуващ logical volume:

1. Пак проверяваме устройствата с "lsblk":

```
[nick@localhost ~]$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                 8:0    0   20G  0 disk
├─sda1                             8:1    0    1G  0 part /boot
├─sda2                             8:2    0   19G  0 part
│ └─cl-root                       253:0    0   17G  0 lvm  /
│ └─cl-swap                       253:1    0    2G  0 lvm  [SWAP]
sdb                                 8:16    0    8G  0 disk
└─nashtagroupa-logicheskibem      253:2    0   6.4G  0 lvm  /home/nick/mountvolume
sdc                                 8:32    0    1G  0 disk
sdd                                 8:48    0    1G  0 disk
sde                                 8:64    0    1G  0 disk
sr0                                 11:0    1 1024M  0 rom
```

2. Създаваме физически обем:

```
[nick@localhost ~]$ sudo pvcreate /dev/sdc
Physical volume "/dev/sdc" successfully created.
```

3. Уголемяваме групата за дадения обем :

```
[nick@localhost ~]$ sudo vgextend nashtagroupa /dev/sdc
Volume group "nashtagroupa" successfully extended
```

4. Уголемяваме логическият обем:

```
[nick@localhost ~]$ sudo lvextend -L 1G /dev/nashtagroupa/logicheskibem
New size given (256 extents) not larger than existing size (1625 extents)
```

5. Променяме размера на файловата система:

```
sudo resize2fs /dev/nashtagroupa/logicheskibem
resize2fs 1.45.4 (23-Sep-2019)
The filesystem is already 1664000 (4k) blocks long. Nothing to do!
```

Най отдолу в LVM Ще имаме физическите обеми на системата → хард дисковете, може и да е само един. /dev/sda, /dev/sdb, /dev/sdc.

След това ще имаме VG ( Групата за обема ) | vg\_base може да е само за един диск или много. Съдържа физическите обеми.



Най отгоре са логическите обеми като например , **lv\_root**, **lv\_var**, **lv\_swap**, **lv\_home** групата, но изсечена на малки порции.

Върху логическите обеми седи файловата система, /, var, swap, home и др.

- **pvs** показва лист с всички физически обеми.
- **vgs** показва групирани обеми
- **lvs** показва логическите обеми

Създаване на обеми и файлови системи:

lsblk команда която се ползва с цел да се покаже лист със блоковите устройства, като хард дисковете:

```
nick@xmachine:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme0n1      259:0    0 953,9G  0 disk
├─nvme0n1p1  259:1    0   512M  0 part /boot/efi
├─nvme0n1p2  259:2    0     1K  0 part
└─nvme0n1p5  259:3    0 953,4G  0 part /
```

Блоково устройство е хардуеър който има записани данни по него.

При определени ситуации е нужно изпълнението на команда partprobe за обновяване на таблиците в ядрото

Кодове на типове дялове – **83 (Линукс файлова система)**, **82 (Swap)**, **5 (extended)** и др.

**fdisk** създава дялове по интерактивен начин **(Преди употреба прочетете man page и погледнете примери онлайн.)**

Промените се извършват в паметта:

- **w** (write) записват се при избор на командата .
- **p** (print) – показване на създадените дялове.
- **n** (new) – създаване на нов дял.
- **d** (delete) – изтриване на дял.
- **t** (type) – смяна на типа на дял.
- **w** (write) – записва таблицата с дяловете на диска и излиза от програмата.
- **q** (quit) – изход от програмата без запис.

Показ на **fdisk**:

1.

```
[nick@localhost ~]$ sudo fdisk /dev/sdd
```

**\*гледайте какво правя\***

parted служи за създаване, модифициране, копиране, оразмеряване и изтриване на MBR и GPT дялове:

- print – показване на създадените дялове.
- mklabel – задава типа таблица на дяловете (gpt, msdos и др.).
- mkpart – създаване на нов дял.
- rm – изтриване на дял.
- set – задава флагове свързани с дяловете.
- quit – изход от програмата.

Показ на **parted**:

Файлови системи в Линукс:

- Ext 2 – дълги години най-използваната Линукс файлова система.
- Ext 3 – журнална версия на ext2.
- Ext 4 – усъвършенствана като производителност и надеждност версия на ext3. Файлова система по-подразбиране на множество Линукс дистрибуции.
- ReiserFS – журнална система, подходяща за управление на малки и много на брой файлове.
- XFS – журнална система с висока производителност и добра скалируемост – по подразбиране в Red Hat Linux Enterprise 7.
- JFS – журнална система разработка на IBM и често използвана за сървърни решения с високи натоварвания.
- MSDOS / VFAT – DOS/Windows 9x файлови системи
- NTFS – Windows NT файлова система – пълна поддръжка при четене и частична за запис.
- NFS – мрежова UNIX файлова система за споделяне на файлове.

- SMB – мрежови протокол разработен за споделяне на файлове в Microsoft Windows и други платформи.

**mkfs -t filesystem\_type (mkfs.ext2, mkfs.ext3, mkfs.jfs, mkfs.reiserfs, mkfs.msdos и др.):**  
Командата която можем да ползваме с цел популизирането на дяла с файлова система.

Веднъж ползвана можем да използваме диска спрямо предназначение.

**mkfs.ext{2,3,4}** – съдържа символични линкове към mke2fs с цел призоваването на файловата система която искаме.

**/etc/mke2fs.conf** – задава параметрите по подразбиране на командата mke2fs

Примери:

<b>mkfs -t ext4 /dev/</b>	Форматира дяла във формата зададен.
<b>mkfs.ext4 /dev/{нашият дял}</b>	Форматира дяла във формата зададен.
<b>mke2fs -t ext4 /dev/{нашият дял}</b>	Форматира дяла във формата зададен.

Как да направим swap в CentOS:

1. Ползвате командата:

- **sudo swapon --show**

```
[nick@localhost ~]$ sudo swapon --show
NAME      TYPE      SIZE USED PRIO
/dev/dm-1 partition 2G   0B   -2
```

2. С командата falldate ще направим нашият swap файл:

- **sudo falldate -l {Размер на swap}G /swapfile**

3. Променяме правата на /swapfile:

- **sudo chmod 600 /swapfile**

4. Нагласяме mkswap /swapfile това ще нагласи Linux swap зоната.

- **sudo mkswap /swapfile**

5. За да направим така, че нашият swap файл да бъде разпознаван от системата, след всеки рестарт трябва да го монтираме във fstab файла.

- Първо копираме нашият fstab в същата директория с цел да имаме стандартният файл.
- След това:

- **sudo vim /etc/fstab**

И добавяме този ред най-оттдолу преглеждайки файла дали има регистрирани други swap структури:

**/swapfile swap swap defaults 0 0**

\*Търсете swap като ключува дума за възможните повторения.

6. Провете със командата:

- **sudo swapon --show**

```
[nick@localhost ~]$ sudo swapon --show
[sudo] password for nick:
NAME      TYPE      SIZE USED PRIO
/dev/dm-1 partition 2G   0B  -2
/swapfile file      4G   0B  -3
```

**\*Моя пример е такъв защото има повторения.**

**Монтирането на файлови системи** става автоматично по време на стартиране (на база съдържанието на /etc/fstab) или ръчно с командата mount.

• /etc/fstab – съдържа информация за всички системи, които трябва да се монтират по време на стартиране във формат:

**[устройство(дял)] [точка на монтиране] [тип файлова система] [опции] [dump] [fsck]**

**mount -t type -o options device mountpoint** – ръчно монтиране на файлова система

**umount device | mountpoint** – демонтира файлова система

**mount -a** – монтира всички системи указани във файла /etc/fstab

**mount** – показва всички монтирани файлови системи

**/etc/mtab** – съдържа информация за монтираните файлови системи, **управлявани** от (u)mount

**/proc/mounts** – съдържа информация за монтираните файлови системи, управлявана от ядрото

**mount --bind** – монтиране на файлова система на няколко места едновременно (

**(Преди опутреба прочетете man page и погледнете примери онлайн.)**

### Разлики между Linux и Windows файлови системи:

- **case sensitivity** – Линукс файловите системи правят разлика между малки и главни букви
- **single root** – всички файлове и директории са под root (/) структурата
- **everything is a file** – в Линукс всичко е файл (включително и директориите)
- **forward slashes** – Линукс използва право наклонени черти (forward slashes) - /home/john/
- **delete or modify open files** – в Линукс можеш да изтриеш или промениш отворени файлове

Настройка и поддръжка на файлова система:

**fsck** - проверка и поправка на проблеми с файловата система

- **e2fsck** - проверка и поправка на проблеми с ext2/3/4 файлови системи.

(Преди опутреба прочетете man page и погледнете примери онлайн.)

- **dumpe2fs** - преглед на информация за ext2/3/4 файловата система.

(Преди опутреба прочетете man page и погледнете примери онлайн.)

- **tune2fs** - задаване на опции за монтиране по подразбиране, настройка на честотата на автоматичния fsck, конвертиране на ext2 в ext3 и ext3 в ext4.

(Преди опутреба прочетете man page и погледнете примери онлайн.)

Домашно:

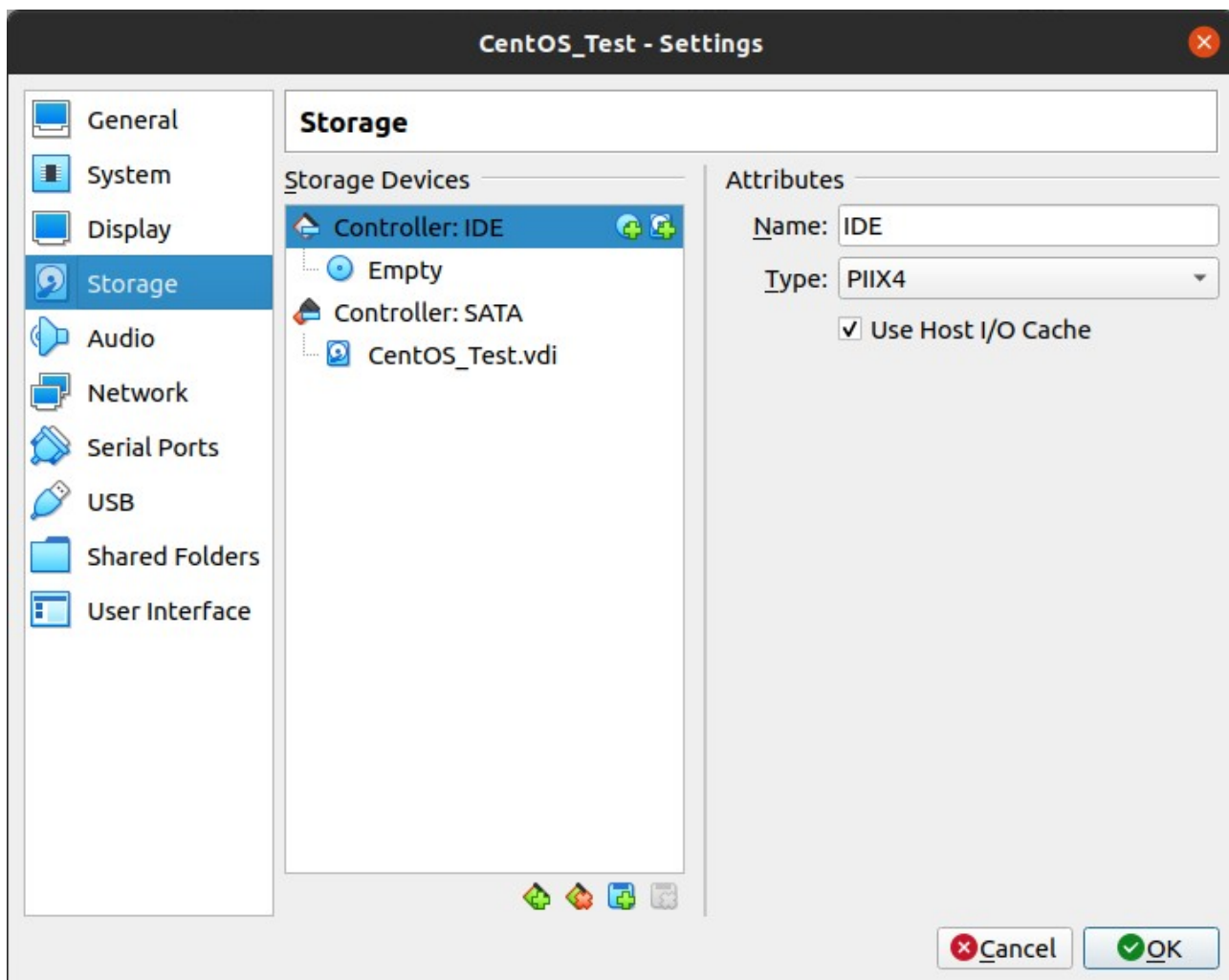
Направете следното:

1. Направете си една виртуална машина със CentOS.



2. След като я изберете с десен бутон изберете настройки или Ctrl+S.
3. Изберете Storage:





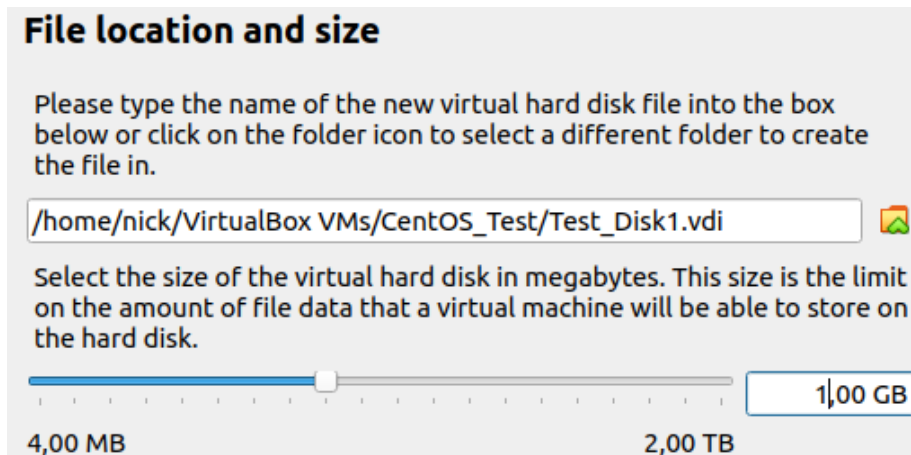
4. При това меню изберете:



5. След това изберете:



6. Наравете 5 диска следвайки тази процедура и следното име за да ви е по лесно, помнете името се задава по-време на тази опция:



**File location and size**

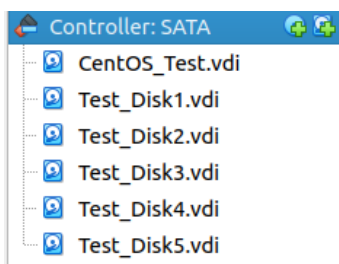
Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.

Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.

4,00 MB 2,00 TB

1,00 GB

7. Повторете няколко пъти докато не направите 5 диска:



8. От тези виртуални дискове във вашата виртуална машина създайте 5 диска и ги монитрайте в /home/ директорията си под папка създадена от вас казваща се mountpoints. Папките могат да съответсват на имената на дисковете. Методиката на създаване на дисковете зависи от вас.