

# Дискове, дялове и swap

08 октомври 2020 г. 13:42

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 4);
- extended дял (максимум 1) – логически дялове;
- максимален дисков размер – 2TB
  - GPT – до 128 дяла;
- не съществуват extended и логически дялове;
- до 9.4 милиарда TB дисков размер

Главни локации във файловата система.

/ --> дъното на директорното дърво 'root'

/var --> променливата локация, log файлове, и динамичното съдържание като например уеб сайтове и мейли

/home --> потребителската папка където се съдържат личните файлове.

/boot --> зареждащата папка, обикновено е на друг партишън. Линукс ядрото и други поддържащи файлове се съдържат там.

/opt --> локация за опционален софтуер, обикновено ползван от 3ти лица. Ентерпрайз приложенията 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 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.

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

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

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

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, 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 logical volumes:

1. pvcreate - инициализация на physical volumes (PV) за използване от LVM.
2. # pvcreate /dev/sda2 /dev/sda5
3. vgcreate - създаване на volume group (VG) и добавяне на PV в нея.
4. # vgcreate volgrpname /dev/sda2 /dev/sda5
5. Опция -s указва размера на PEs.
6. lvcreate - създаване на logical volumes в дадена VG. Всеки LV притежава уникално име, което може да бъде указано (опция -n) или автоматично генерирано.
7. # lvcreate -L 650M -n lvname volgrpname
8. Опция -L указва размера на създавания LV.
9. Създаване на файлова система на LV - mkfs -t ext4 /dev/volgrpname/lvname
10. Монтиране на logical volumes - mount /dev/volgrpname/lvname /dir

Управление на VG и LV:

1. Преглед - pvdisplay, vgdisplay, lvdisplay, lvmdiskscan, pvs, vgs, lvs

2. Оразмеряване - `vgreduce`, `vgextend`, `pvresize`, `lvreduce`, `lvextend`, `lvresize`,
3. `resize2fs`, `xfs_growfs`
4. Добавяне на нов физически диск към съществуващ logical volume:
5. `#pvcreate /dev/sdb`
6. `#vgextend vg01 /dev/sdb`
7. `#lvextend -L 500G /dev/vg01/databases`
8. `#resize2fs /dev/vg01/databases`

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 на 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 групираният 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.

Блоково устройство е хардуер 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

Промените се извършват в паметта – записват се при избор на командата `w` (write)

`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

Как изглежда:

```
# fdisk /dev/sda
```

Command (m for help): `n`

Partition type

`p` primary (1 primary, 1 extended, 2 free)

`l` logical (numbered from 5)

Select (default `p`): `p`

Partition number (3,4, default 3): (press ENTER)  
First sector (18571264-22765679, default 18571264): (press ENTER)  
Last sector, +sectors or +size{K,M,G,T,P}(18571264-18573309,default  
18573309): +500K  
Created a new partition 3 of type 'Linux' and of size 500 KiB.

Command (m for help): t  
Partition number (1-3,5, default 5): 3  
Partition type (type L to list all types): 82  
Changed type of partition 'Linux' to 'Linux swap / Solaris'.  
Command (m for help): p  
Disk /dev/sda: 10,9 GiB, 11656028160 bytes, 22765680 sectors  
Device Boot Start End Sectors Size Id Type  
/dev/sda1 \* 2048 18571263 18569216 8,9G 83 Linux  
/dev/sda2 18573310 22763519 4190210 2G 5 Extended  
/dev/sda3 18571264 18572263 1000 500K 82 Linux swap / Solaris  
/dev/sda5 18573312 22763519 4190208 2G 82 Linux swap / Solaris

Command (m for help): d  
Partition number (1-3,5, default 5): 3  
Partition 3 has been deleted.  
Command (m for help): w  
The partition table has been altered.  
Calling ioctl() to re-read partition table.  
Re-reading the partition table failed.: Device or resource busy  
The kernel still uses the old table. The new table will be used at  
the next reboot or after you run partprobe(8) or kpartx(8).  
# partprobe /dev/sda

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

# parted

GNU Parted 3.2

Using /dev/sda

Welcome to GNU Parted! Type 'help' to view a list of  
commands.

(parted) rm 3

(parted) mkpart primary linux-swap 9508MB 9509MB

(parted) print

Disk /dev/sda: 11,7GB

Partition Table: msdos

Disk Flags:

Number Start End Size Type File system Flags

1 1049kB 9508MB 9507MB primary ext4 boot

3 9508MB 9509MB 513kB primary linux-swaps(v1) lba

2 9510MB 11,7GB 2145MB extended

5 9510MB 11,7GB 2145MB logical linux-swaps(v1)

(parted) quit

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 – обвивка на програмите създаващи съответните

файлови системи (mkfs.ext2, mkfs.ext3, mkfs.jfs, mkfs.reiserfs, mkfs.msdos и др.)

mkfs.ext{2,3,4} – символични линкове към mke2fs

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

Примери:

```
# mkfs -t ext4 /dev/sda2
```

```
# mkfs.ext4 /dev/sda2
```

```
# mke2fs -t ext4 /dev/sda2
```

Swap примери:

```
# swapon /dev/sda3
```

```
# swapon --show
```

```
NAME TYPE SIZE USED PRIO
```

```
/dev/sda5 partition 2G 31,7M -
```

```
/dev/sda3 partition 496K 0B -2
```

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, управлявана от `(u)mount`

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

`mount --bind` – монтиране на файлова система на няколко места едновременно

Монтиране на файлови системи -

примери

```
# cat /etc/fstab
```

```
UUID=2cee2256-32f3-4d9b-9a4d-43ad8bfaaff2 / ext4 errors=remount-ro 0 1
```

```
UUID=072cf7d2-0787-48a5-b8b4-6d9dc904195b none swap sw 0 0
```

```
# umount /dev/sda3
```

```
# mount /dev/sda3
```

```
# mount -t ext4 /dev/sda3 /mnt/
```

```
# mount --bind /var/named /chroot/named/var/named
```

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
- `dumpe2fs` - преглед на информация за ext2/3/4 файловата система
- `tune2fs` - задаване на опции за монтиране по подразбиране, настройка на честотата на автоматичния `fsck`, конвертиране на ext2 в ext3 и ext3 в ext4

e2label - задаване на етикети за означаване на ext2/3/4 файлови системи

- debugfs - преглед и промяна на състоянието на елементи от ext2/3/4 файлови системи
- xfs\_repair - поправка на проблеми с XFS файлови системи
- xfs\_info - показва информация за XFS файлови системи
- xfs\_metadump - копира метаданните на XFS файлова система във файл

Конфигуриране на дискови квоти:

- Дисковите квоти (disk quotas) са полезни за ограничаване на максималния брой
- блокове или inodes, които потребители и групи могат да използват.
- Конфигурацията на дискови квоти включва 4 етапа:

1. Монтиране на файловата система с опция usrquota или grpquota:

```
# mount -o remount, usrquota /home
```

2. Създаване на база данни на квотите, съдържаща текущото потребление на файловата система и зададените лимити. Базата данни е изградена от два файла aquota.user и aquota.group, които се намират в root директорията на съответната файлова система. Първоначалната им инициализация става чрез командата quotacheck:

```
# quotacheck -cugm /home
```

3. Активиране / деактивиране на дисковите квоти с обръщение към ядрото:

```
# quotaon / quotaoff /home
```

4. Задаване на конкретни квоти на потребител или група - setquota (неинтерактивно) и edquota (интерактивно чрез текстови редактор).

Типове лимити - файлови/inode лимити (hard и soft), блокови лимити (hard и soft), гратисен период (grace period).

Hard лимити - абсолютни ограничения, които не могат да бъдат надвишавани.

Soft лимити - могат да бъдат надвишавани от потребителя за определен гратисен период от време. След изтичане на този период действат като hard лимити.

```
# setquota test 50M 100M 0 0 /home # 50MB soft и 100MB hard лимити
```

```
# edquota test
```

Преглед на дискови квоти:

quota - показва информация за квотите на база потребител или група:

```
# quota test
```

- repquota - показва информация за квотите на база файлова система

```
# repquota /tmp
```

Единствено root може да преглежда информация за квотите на други потребители.