**Файловые системы**

* **Ext4** - журналируемая файловая система, используемая в ОС на ядре Linux. Основана на файловой системе Ext3, но отличается тем, что в ней представлен механизм записи файлов в непрерывные участки блоков (екстенты), уменьшающий фрагментацию и повышающий производительность. В Ubuntu, начиная с версии 9.10, эта файловая система используется по умолчанию при автоматическом разбиении диска инсталлятором.
* **Ext3** - журналируемая файловая система, используемая в ОС на ядре Linux. Является файловой системой по умолчанию во многих дистрибутивах. Основана на Ext2, но отличается тем, что в ней есть журналирование, то есть в ней предусмотрена запись некоторых данных, позволяющих восстановить файловую систему при сбоях в работе компьютера.
* **Ext2** - файловая система, используемая в операционных системах на ядре Linux. Достаточно быстра для того, чтобы служить эталоном в тестах производительности файловых систем. Она не является журналируемой файловой системой и это её главный недостаток.
* **BTRFS** - Достаточно новая универсальная ФС, используемая в операционных системах на ядре Linux. Ее особенностями являются: индексное хранение структур данных (в В-деревьях), возможность создания снепшотов, и многие другие интересные вещи.
* **XFS** - высокопроизводительная журналируемая файловая система. Распределение дискового пространства - екстентами, храниение каталогов в B-деревьях. Автоматическая аллокация и высвобождение I-node. Дефрагментируется «на лету». Невозможно уменьшить размер существующей файловой системы. При сбое питания во время записи возможна потеря данных (хотя этот недостаток нельзя относить к одной только XFS, он свойственен любой журналируемой ФС, но, вместе с тем, XFS, по умолчанию, достаточно активно использует буферы в памяти).
* **Fat16** - файловая система, сейчас широко используемая в картах памяти фотоаппаратов и других устройств.
* **Fat32** - файловая система основанная на Fat16. Cоздана, чтобы преодолеть ограничения на размер тома в Fat16.
* **NTFS** - файловая система для семейства операционных систем Microsoft Windows. Поддержка в Ubuntu осуществляется специальным драйвером - NTFS-3G.
* **HFS** - файловая система, разработанная Apple Inc. для использования на компьютерах, работающих под управлением операционной системы Mac OS.
* **HSF+** - файловая система, разработанная Apple Inc. для замены HFS. Является улучшенной версией HFS, с поддержкой файлов большого размера и использует кодировку Unicode для имён файлов и папок.
* **JFS** - журналируемая файловая система. В отличие от Ext3, в которую добавили поддержку журналирования, JFS изначально была журналируемой. На момент выхода в свет JFS была самой производительной из существовавших файловых систем. На текущий момент сохраняет за собой одно из лидирующих мест по этому показателю.
* **SWAP** - раздел жёсткого диска, предназначенная для виртуальной памяти (файла подкачки).
* **ReiserFS** - журналируемая файловая система, разработанная специально для Linux. Обычно под словом ReiserFS понимают третью версию (последняя — 3.6.21), а четвёртую называют Reiser4. В настоящий момент разработка Reiser3 прекращена.
* **Reiser4** - журналируемая файловая система ReiserFS (4-я версия), разработанная специально для   Linux. Одна из самых быстрых файловых систем для Linux (с включённым плагином-архиватором - самая быстрая).
* **UFS** - файловая система, созданная для операционных систем семейства BSD. Linux поддерживает UFS на уровне чтения, но не имеет полной поддержки для записи UFS. Родной Linux ext2 создан по подобию UFS.

**fdisk –** консольная утилита для разбивки дисков. Создает/удаляет разделы, но не форматирует их

Работает в интерактивном режиме

**sudo fdisk -l** – посмотреть какие диски есть в системе

**sudo fdisk -l /dev/**sdx – просмотр информации о конкретном диске

**sudo fdisk /dev/**sdx – перейти в оснастку редактирования диска

* **m** -краткая справка
* **p** - текущий список дисковых разделов с указанием их типа и размер
* **o** – создать MBR таблицу разделов
* **n** – создать новый раздел
  + При назначении last sector можем использовать **+**[количество]**G** что указать сколько гигабайт будет раздел
* **d** – удалить раздел
* **w** – записать изменения
* **q** - выход

**parted** - универсальное средство для работы не только с дисковыми разделами, но и с файловыми системами. Позволяет не только выполнить разметку диска, но и создать на разделах файловые системы, а также осуществляет проверку их целостности, удаление, перемещение, копирование и изменение размера разделов существующих.

В отличии от fdisk, сразу записы вает изменения на диск после выполнения команды

Работает в интерактивном и командном режимах

**sudo parted /dev/**sdx – запуск parted в интерактивном режиме

* **help** – посмотреть список команд
* **select /dev/**sdx – выбрать другой диск
* **print** – получить информацию о разделах диска
* **mklabel msdos|gtp** – создать таблицу разделов (msdos - MBR)
* **mkpart [part-type fs-type name start end]** – создать раздел
  + **part-type** - primary|extended|logical указывается для MBR
  + **fa-type** - файломая система, см [Форматирование разделов](https://wiki.a1qa.com/pages/viewpage.action?pageId=125994603)
  + **name** - имя раздела, указывается для GPT
  + **start** - начало раздела, может указываться в [unit](http://www.gnu.org/software/parted/manual/parted.html#unit) или в %
  + **end** - конец раздела, может указываться в [unit](http://www.gnu.org/software/parted/manual/parted.html#unit) или в %

Если не задавать параметры part-type, fs-type, name, start, end, то они будут запрошены

* **quit** - выход

**Форматирование разделов**

Отформатировать раздел в Linux можно следующим образом

**mkfs -t *<fstype> <device>***

или

**mkfs.*<fstype> <device>***

Где <fstype>:

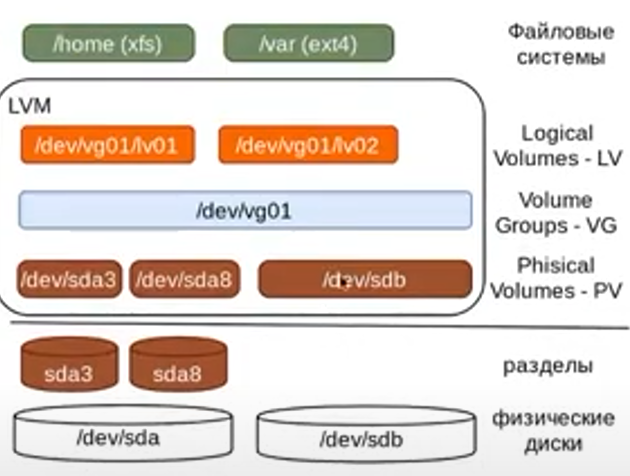
* reiserfs
* ext3
* ext4
* jfs
* xfs
* vfat
* ntfs
* и т.д.

**Volume Manager (группа томов)**

Linux Volume Manager (LVM) — система управления томами, которая позволяет создавать поверх физических разделов (или даже неразбитых винчестеров) логические тома, которые в самой системе будут видны как обычные блочные устройства с данными

Преимущества LVM:

* Возможность изменения размера.
* Размер файловых систем логических томов не ограничивается одним диском, так как том  может располагаться на разных дисках и разделах.
* Изменение размера.
* С помощью простых команд можно изменить размер логических томов без необходимости переформатирования или переразбиения физических дисков в их основе.
* Перемещение данных в активной системе.
* С целью создания новых, более быстрых и устойчивых под систем хранения д опускается  перемещать данные в активной системе. Данные можно переносить, даже если к дискам выполняется обращение. Например, можно на ход у освободить заменяемый диск передегудалением из набора.
* Простота присвоения имен устройствам.
* Логические тома могут объединяться в группы для облегчения управления. Группам могут присваиваться произвольные имена.
* Чередование дисков.
* Для повышения пропускной способности можно создать логический том с чередованием
* Данных на нескольких дисках.
* Зеркалирование томов.
* Использование логических томов позволяет создать зеркальные копии данных.
* Снимки томов.
* Снимки томов представляют собой их точные копии и могут использоваться для тестирования, не повергая риску исходный том



LVM тома создаются в группе, которая, в свою очередь разворачивается на нескольких разделах или дисках целиком.

Есть 3 уровня абстракции:

1. **PV (Physical Volume)** — физические тома (это могут быть разделы или целые «неразбитые» диски)

2. **VG (Volume Group)** — группа томов (объединяем физические тома (PV) в группу, создаём единый диск, который будем дальше разбивать так, как нам хочется)

3. **LV (Logical Volume)** — логические разделы, собственно раздел нашего нового «единого диска», который мы потом форматируем и используем как обычный раздел, обычного жёсткого диска.

**lvm** – запуск Linux Volume Manager

* **help** – вывести список всех команд
* **pvcreate /dev/**sdx ; **pvcreate /dev**/sdx – указать физические устройства, на которых создаем группу томов
* **pvremove /dev/sd**x – удалить LVM labels
* **pvdisplay /dev/sd**x – показать атрибуты физических дисков
* **vgcreate** vg\_name **/dev/sd**x **/dev/sd**x2– создать группу томов
* **vgremove** vg\_name– удалить группу томов
* **vgdisplay** vg\_name– показать информацию по группе томов
* **vgextend** vg\_name– добавить физический диск в группу томов
* **lvcreate** -**n** name-**L** disk\_sizevg\_name – создать логический диск
  + -**L [K, M, G]** – Kilobytes, Megobytes, Gigobytes
  + **-l <N>%FREE** – позволяет указать объем в процентах
* **lvremove** name– удалить логический том
* **lvdisplay** name– показать информацию о логическом диске
* **lvextend** [params] – расширить логический диск

**Создание разделов**

**1)** **Указываем физические устройства, на которых создаем группу томов:**

|  |
| --- |
| pvcreate /dev/sda3; pvcreate /dev/sdb |

**2) Создаем группу томов:**

|  |
| --- |
| vgcreate vg1 /dev/sda3 /dev/sdb |

3**) Создаем разделы:**

|  |
| --- |
| lvcreate -n lv1 -L 20G vg1  lvcreate -n lv2 -l 100%FREE vg1 |

После того как разделы созданы им будут соответствовать файлы устройств **/dev/<имя группы>/<имя тома>**, то есть в нашем примере это будет /dev/vg1/lv1 и /dev/vg1/lv2. Остается только их [отформатировать](https://wiki.a1qa.com/pages/viewpage.action?pageId=125994603) и [монтировать](https://wiki.a1qa.com/pages/viewpage.action?pageId=126618450).

**Добавление томов в группу**

**1) Указываем физические устройства, которые хотим добавить:**

|  |
| --- |
| pvcreate /dev/sdc |

**2) Добавляем их в группу томов:**

|  |
| --- |
| vgextend vg1 /dev/sdc |

**3)**[**Создаем разделы**](https://wiki.a1qa.com/pages/viewpage.action?pageId=125994966#id-5.LinuxVolumeManager(LVM)-lvmcreate)**или увеличиваем размер существующего:**

|  |
| --- |
| lvextend -L +15g vg1/lv2 |

**4) Расширяем файловую систему чтобы можно было использовать новое пространство:**

|  |
| --- |
| resize2fs /dev/vg1/lv2 |

**Удаление томов из группы**

**1) Переносим данные с тома на другие диски:**

|  |
| --- |
| pvmove /dev/sda3 |

**2) Удаляем его из группы:**

|  |
| --- |
| vgreduce vg1 /dev/sda3 |

**3) Удаляем физический том из LVM:**

|  |
| --- |
| pvremove /dev/sda3 |

**Работа со снапшотами**

**Создаем снапшот**

|  |
| --- |
| lvcreate -s -n snap -L 20g vg1/lv1 |

**Откатывается до снапшота**

|  |
| --- |
| lvconvert --merge vg1/snap |

 Когда мы создаем снапшот, мы создаем новый логический раздел, который изначально не использует дисковое пространство. По мере того как на оригинальном разделе вносятся изменения, раздел-снапшот заполняется данным, то есть чем больше изменений вносим в оригинальный раздел, тем больше заполняется раздел-снапшот. Если свободное место на снапшоте заканчивается, он становится больше непригодным и с ним больше нельзя работать.

**Монтирование разделов**

**Монтирование** - это привязка блочного устройства (жесткого диска, флешки, cd-rom т.д.) к определенному каталогу, который называют точкой монтирования

**mount *<device****>* *[*-t *<fstab>]* ***<directory>***

где **<device>** - устройство

**<fstab>** - файловая система, например ext4, ntfs

**<directory>** - точка монтирования. Каталог должен быть предварительно создан

Другие полезные флаги mount:

**-a** - cмонтировать все файловые системы, перечисленные в файле /etc/fstab. Исключение составляют помеченные как noauto, перечисленные после опции -t и уже смонтированные

**-f** - монтировать поврежденный раздел (опасно!), или форсировать отмену всех запросов на запись при изменении режима монтирования с "чтение-запись" на "только чтение"

**-r** - монтировать файловую систему в режиме "только для чтения". То же самое, что и указание аргумента ro для опции -o

**-v** - выдавать более подробную информацию

**-w** - монтировать файловую систему в режиме "чтение-запись". То же самое, что и указание аргумента rw для опции -o

**umount *<device>/<derectory>*** *-* демонтировать

**Пример:**

**sudo mount /deb/**sdb2 **/mnt**

Если нужно, чтобы раздел монтировался при загрузке системы, нужно отредактировать файл **/etc/fstab**

Структура фала:

*устройство* */точка-монтирования* *тип файловой системы* *опции* *частота дампов* *порядок проверки*

***устройство*** - Имя устройства (UUID или путь)

***точка монтирования*** - Каталог, куда следует смонтировать файловую систему

***тип файловой системы*** - Тип файловой системы, который передается программе mount.

***опции*** - По умолчанию используются defaults, rw, для монтирования файловой системы в режиме "чтение-запись", или ro, для режима "только чтение", за которыми могут следовать и другие опции. Довольно часто используется опция noauto, чтобы не монтировать автоматически файловые системы в процессе загрузки.

***частота дампов*** - Используется утилитой dump для определения файловых систем, с которых необходимо периодически снимать специальные архивные копии. При отсутствии этого параметра принимается равным нулю

***порядок проверки*** - Определяет порядок, в котором следует проверять файловые системы (чаще всего, в случае некорректного размонтирования или внезапной перезагрузки системы). Если файловую системы не нужно проверять, этот параметр должен быть установлен в ноль. Для корневой файловой системы (которая должна быть проверена в первую очередь) установите его в 1. Для всех остальных - 2 или больше. Если две или более файловые системы имеют одинаковое значение passno, fsck попытается проверять их параллельно (если, конечно, это возможно физически).

**Пример:**

/dev/vg1/lvm1-1 /lvm1-1 ext3 defaults 0 2

 Посмотреть список всех монтированных разделов можно в файле **/etc/mtab.**