
Предисловие

На этом занятии мы поговорим о:

- разделах и файловых системах;
- типах файловых систем;
- организация хранения данных;
- специальных файловых системах.

По итогу занятия вы узнаете как устроены файловые системы, какие из них встречаются в Linux и какие инструменты можно использовать для их настройки.

План занятия

1. [Предисловие](#)
2. [Разделы и файловые системы](#)
3. [Файловые системы](#)
4. [Специальные файловые системы](#)
5. [Сетевые и распределенные файловые системы](#)
6. [Монтирование файловых систем](#)
7. [Хранения данных. Inodes.](#)
8. [Итоги](#)
9. [Домашнее задание](#)



Разделы и файловые системы

Разделы и файловые системы

- Почему недостаточно разделов?
- Для чего нужны файловые системы?



Файловая система

Часть операционной системы, которая **устанавливает физическую и логическую структуру файлов** на разделе или диске, **управляет созданием и изменением файлов** и сопутствующих данных.

В Linux на каждый раздел можно установить свою ФС, которая отвечает за порядок и способ организации информации.

От выбора файловой системы зависят:

- скорость работы с файлами;
- их сохранность;
- скорость записи;
- размер файлов.

Функции файловых систем

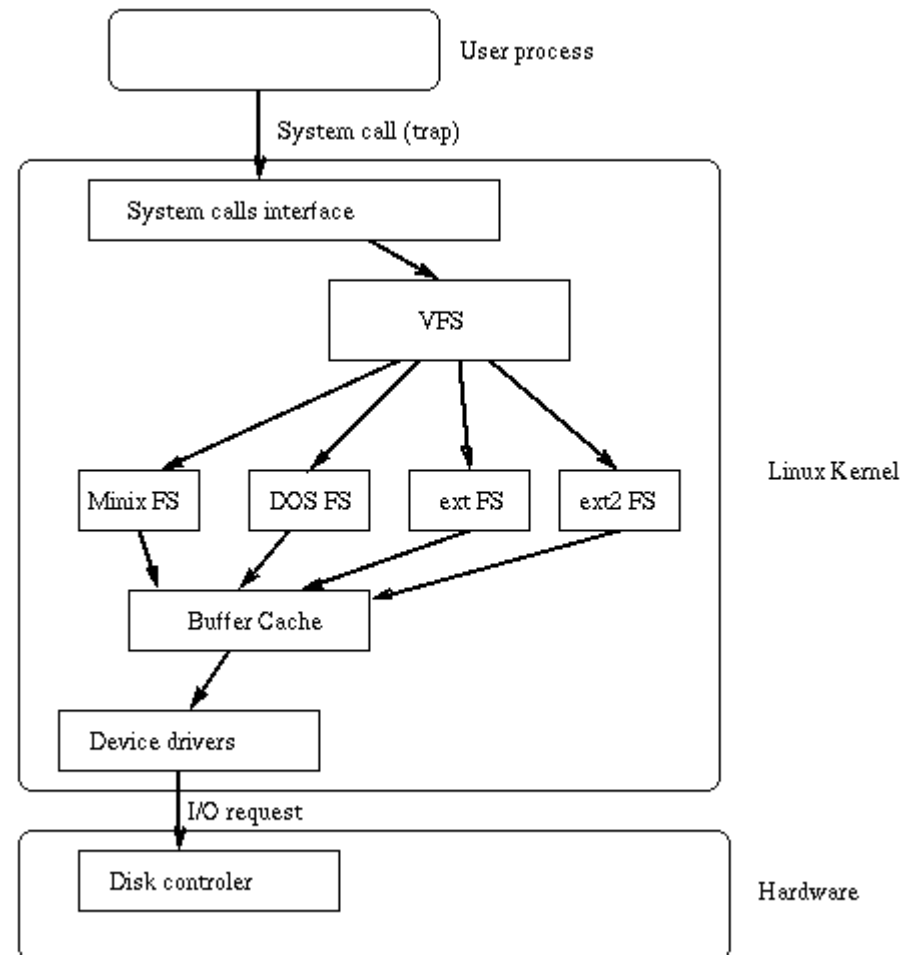
- размещение и упорядочивание на носителе данных в виде файлов;
- создание, чтение и удаление файлов;
- назначение и изменение атрибутов файлов;
- защита файлов при системном сбое;
- защита файлов от несанкционированного доступа;
- поиск файлов.

Разделы файловые системы

- размещение и упорядочивание на носителе данных в виде файлов;
- создание, чтение и удаление файлов;
- назначение и изменение атрибутов файлов;
- защита файлов при системном сбое;
- защита файлов от несанкционированного доступа;
- поиск файлов.

Виртуальная файловая система

С помощью виртуальной файловой системы(VFS) происходит совместная работа ядра и приложений, установленных в системе. VFS позволяет пользователю работать, не учитывая особенности каждой конкретной ФС.



Команды Linux для работы с разделами

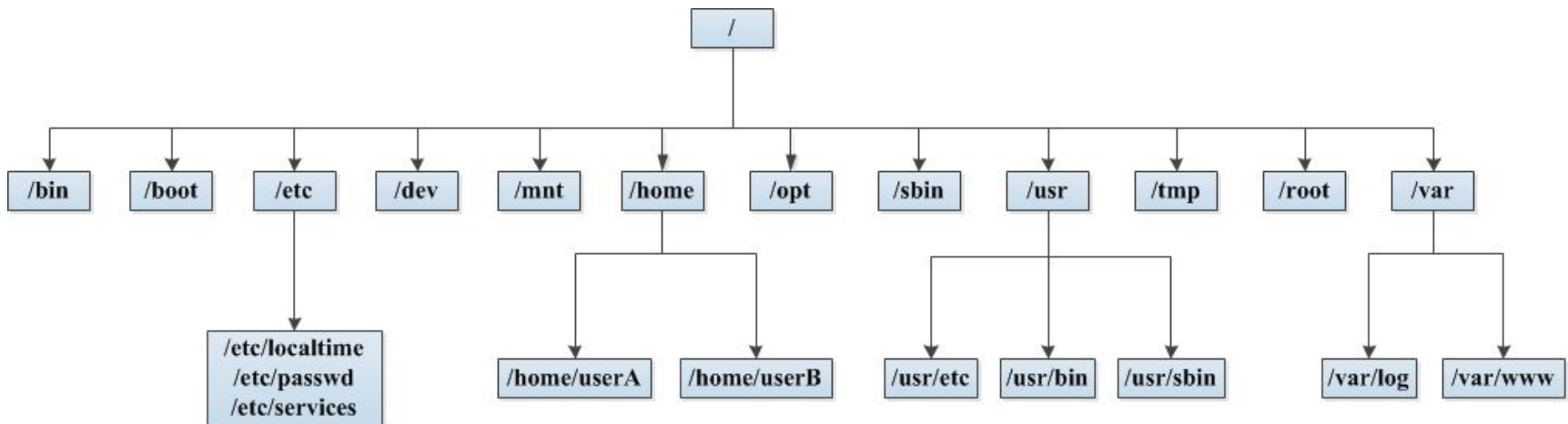
- **cat /proc/filesystems** – вывести список файловых систем, которые поддерживаются ядром
- **fsck** – утилита, с помощью которой можно проверить ФС на ошибки.
- **file -s /dev/sda1** - выводит тип файловой системы
- **df -T** - выводит тип файловой системы.



Файловые системы

Иерархия каталогов

Filesystem Hierarchy Standard (FHS - стандарт иерархии файловых систем) - стандарт на структуру каталогов для UNIX-подобных ОС



Сравнение файловых систем

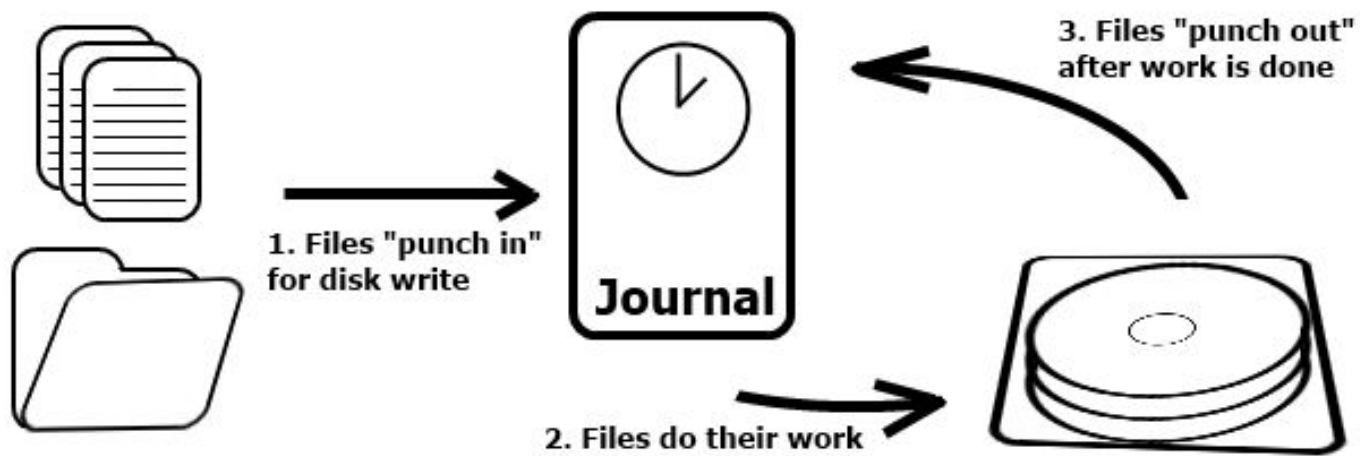
	журналирование	экстенты	cow
ext2	-	-	-
ext3	+	-	-
ext4	+	+	-
xfs	+	+	-
btrfs	+	+	+

Журналирование

- При использовании **журналирования** файл сначала записывается в журнал («лог»).
- После этого файл записывается на жесткий диск, а потом удаляется из журнала, после чего операция записи считается завершённой.
- Если во время записи выключилось питание, то после включения системы файловая система может проверить журнал и найти незавершённые операции.

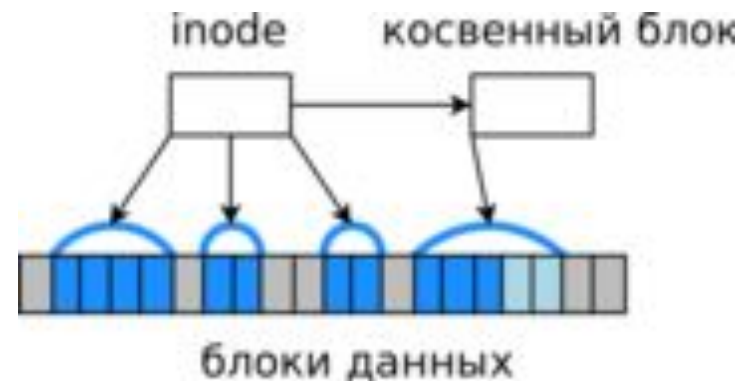
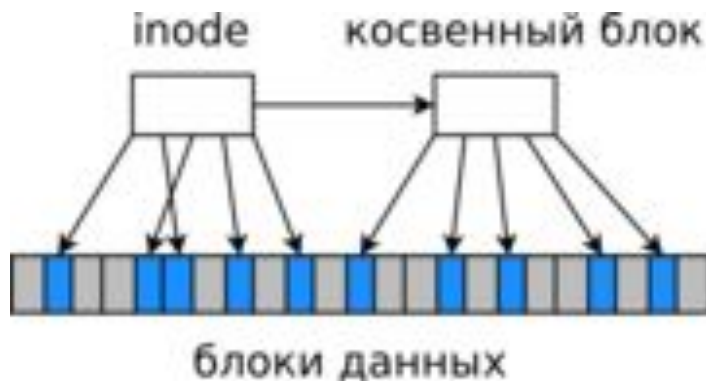
Журналирование в том или ином виде применяется практически во всех современных файловых системах.

Журналирование



Extent

В традиционных файловых системах заголовки(inode) указывают на отдельные блоки . В файловых системах с поддержкой экстентов используются указатели не на блоки, а на экстенты - непрерывные области носителя информации.





COW

COW(Copy-On-Write - копирование при записи) - механизм для оптимизации различных процессов в операционных системах. Идея подхода copy-on-write заключается в том, что при чтении области данных используется общая копия, в случае изменения данных — создается новая копия.

FAT

FAT (File Allocation Table, таблица размещения файлов) – файловая система, которая использовалась в операционной системе DOS.

В файловой системе FAT дисковое пространство логического раздела делится на три области: зарезервированная, системная и область данных .

Загрузочная запись	FAT	FAT (копия)	Корневой каталог	Область файлов
--------------------	-----	-------------	------------------	----------------

Команды для создания FAT

- `yum install dosfstools;`
- `fdisk /dev/sd*;`
- `mkfs.vfat -F 32 -n MyDrive /dev/sd*;`
- `fatresize -s /dev/sd*.`

NTFS

NTFS (new technology file system, «файловая система новой технологии») — стандартная файловая система для семейства операционных систем Windows NT.

Для работы с NTFS может потребоваться установка пакетов:

- **yum install ntfs-3g** — установка необходимых пакетов;
- **mount -t ntfs-3g /dev/sdb1 /mnt** — монтирование NTFS в Linux.

Ext2

Ext2 — файловая система с хорошими показателями **скорости**, однако это было достигнуто за счет **отказа от журналирования**.

Команда для создания ext2 на разделе */dev/hdb1*:

```
mkfs -t ext2 /dev/hdb1
```

Ext4

Ext4 — является развитием Ext2 и Ext3.

Самая популярная, “файловая система по умолчанию” в Linux.

В отличие от Ext3 введен механизм пространственной (extent) записи файлов, **уменьшающего фрагментацию и повышающего производительность.**

Команда для создания ext4 на разделе */dev/hdb1*:

```
mkfs -t ext4 /dev/hdb1
```

XFS

XFS - высокопроизводительная 64-битная журналируемая файловая система.

Особенности XFS:

- Изначально рассчитана для использования на дисках большого объема, более 2 Тбайт;
- Восстановления потерянных файлов в XFS затруднительно;
- Уменьшить размер раздела файловой системы невозможно;
- журналирование метаданных;
- дефрагментация на лету.

Команды для работы с XFS

- **mkfs.xfs /dev/sda1** - создание файловой системы
- **xfs_info /dev/sda1** - выводит информацию про метаданные файловой системы.
- **xfs_growfs / -d** - увеличивает файловую систему на все доступное пространство
- **xfs_check /dev/sdb1, xfs_repair /dev/sdb1** - проверяет файловую систему на ошибки



Btrfs

Btrfs (B-tree FS) — файловая система для Linux, основанная на структурах B-деревьев и работающая по принципу «копирование при записи» (copy-on-write).

Особенности Btrfs

- дедупликация данных;
- снапшоты и сабвоリュームы;
- сжатие данных;
- copy on write;
- возможность объединить несколько накопителей в единую файловую систему (LVM не нужен);
- нежурналируемая;
- поддержка преобразования из ext4 и ext3 и обратно;
- создает RAID на уровне файловой системы.

Команды для работы с Btrfs

- **mkfs.btrfs /dev/sdc -L single_drive** - создание ФС на одном диске
- **mkfs.btrfs /dev/sdc /dev/sdd -L double_drive** создание ФС на двух дисках
- **sudo mkfs.btrfs /dev/sdc /dev/sdd -d raid1 -m raid1 -L raid1_drive** создать рейд средствами btrfs
- **btrfs filesystem df /** - получить информацию о ФС
- **btrfs filesystem resize -2g /mnt** - изменить размер тома в реальном времени



Специальные файловые системы



sysfs

sysfs — специальная ФС, которая предоставляет детализированную информацию о работе ядра, присутствующих в системе устройствах и драйверах пользователю.

procfs

Procfs — специальная файловая система, используемая в UNIX-подобных ОС. По умолчанию смонтирована в папку /proc. Предлагает моментальный снимок состояния ядра и процессов.

➡ В этой ФС храниться подробная информация об ОС, включая ядро, процессы и параметры конфигурации.

➡ Пронумерованные директории соответствуют PID запущенных в системе процессов.

tmpfs

tmpfs — это файловая система, которая располагается в оперативной памяти. Вы можете создать блочное устройство нужного размера в ОП, затем смонтировать его в директорию, и вы можете работать с этим устройством как с любым другим блочным устройством. За исключением того, что после перезагрузки данные очистятся.



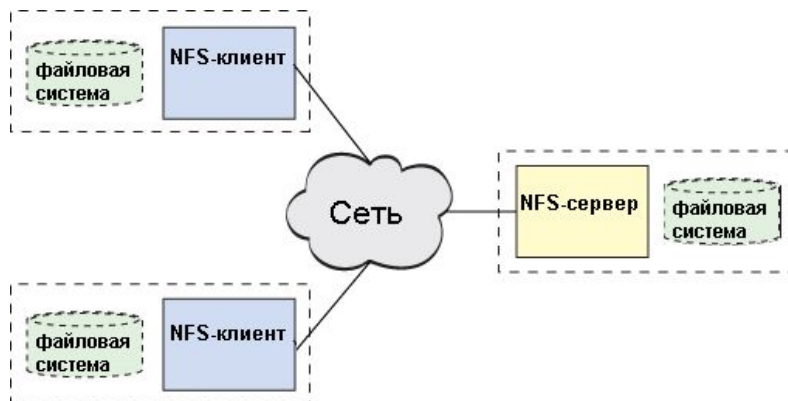
Сетевые/Распределенные/ Кластерные файловые системы

NFS

NFS (Network File System, сетевая файловая система) — протокол сетевого доступа к файловым системам. Позволяет пользователям подключать удаленные сетевые каталоги к своей системе и передавать файлы между серверами.

NFS:

- клиент-серверная архитектура;
- для подключения используется команда `mount`.



SMB/CIFS

SMB(Server Mesage Block), Common Internet File System (CIFS) (Единая файловая система Интернета)— это сетевой протокол прикладного уровня для удалённого доступа к файлам, принтерам и другим сетевым ресурсам, а также для межпроцессного взаимодействия.

Samba - пакет программ, которые позволяют обращаться к сетевым дискам и принтерам на различных операционных системах по протоколу SMB/CIFS

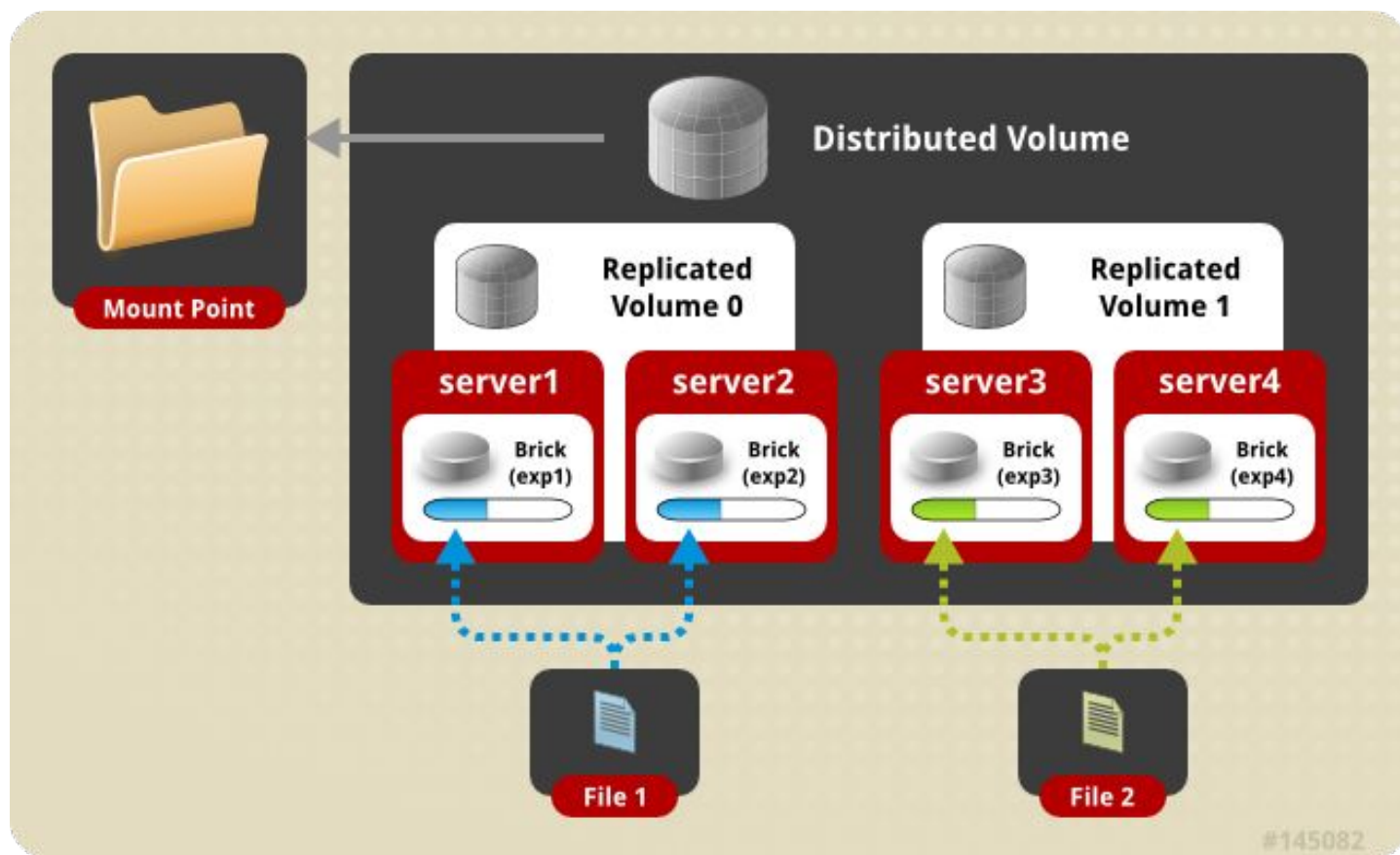
GlusterFS


GlusterFS — это распределенная, параллельная, линейно масштабируемая файловая система с возможностью защиты от сбоев.

Позволяет объединить:

- хранилища данных, расположенные на разных серверах;
- хранилища, расположенные на одном сервере в одну сетевую файловую систему.

Пример реализации GlusterFS





Монтирование файловых систем

Файл `fstab`

Файл `fstab (/etc/fstab)` — конфигурационный файл, содержащий инструкции по монтированию блочных устройств, NFS-ресурсов и псевдо-файловых систем в пространство файловых имен и областей подкачки страниц.

Пример записи в файле `fstab`:

```
UUID=7211313d-8b5e-4b96-be63-5dbbeb97f590 /          ext4  errors=remount-ro 0      1
```

Определение полей fstab

- **что монтируем** — некоторое блочное устройство, которое должно быть примонтировано;
- **куда монтируем** — точка монтирования — путь в корневой файловой системе к каталогу в который будет смонтировано устройство;
- **тип** файловой системы монтируемого раздела;
- **опции** монтирования файловой системы;
- **индикатор** необходимости делать **резервную копию** (как правило не используется и равно 0);
- **порядок проверки раздела** (0 — не проверять, 1 — устанавливается для корня, 2 — для остальных разделов).

Опции монтирования

Некоторые опции монтирования могут быть специфичны для конкретной реализации ФС. Справка по общим опциям монтирования - **man mount**.

- **noauto** — предотвращает монтирование при загрузке;
- **ro** или **rw** — задает можно ли писать в раздел или он будет доступен только для чтения;
- **noatime** - отключает создание временных меток на разделе;
- **опции** монтирования файловой системы;
- **defaults** - использует следующий набор меток: rw, suid, dev, exes, auto, nouser, async;

Автоматическое монтирование autofs

Может быть реализовано с помощью пакета **autofs** и позволяет автоматически подключать различные ресурсы (сетевые устройства, жесткие диски) в начале их использования и отключать при прекращении использования.

Команды для начала работы с автмонтированием:

- **yum install autofs** – установка пакета;
- **man auto.master** – получение справки по пакету;

Автоматическое монтирование systemd

Автомонтирование также может быть реализовано с помощью unit-файлов systemd.

Команды для начала работы с systemd-монтированием:

- **systemctl list-units -t mount --all** – вывести состояние всех описанных в systemd точек монтирования;
- **systemctl edit --full boot.mount** - открыть на редактирование unit-файл.

Команды Linux для работы с mount

- **mount -t ext4 -o noexec /dev/sdb6 /mnt** – монтировать фс ext4 без возможности запускать исполняемые файлы на разделе;
- **mount --uuid="b386d309-05c1-42c8-8364-8d37270b69e0" /mnt** – смонтировать раздел используя UUID;
- **mount --label="home" /mnt/** – монтировать раздел используя лейбл;
- **mount** – вывести список всех смонтированных устройств.
- **umount /mnt** - отмонтировать раздел, смонтированный по указанному пути.



Хранение данных. Inodes.

Все в Linux есть файл

Типы файлов в Linux

Типы файлов		Назначение
Обычные файлы	—	Хранение символьных и двоичных данных
Каталоги	d	Организация доступа к файлам
Символьные ссылки	l	Предоставление доступа к файлам, расположенных на любых носителях
Блочные устройства	b	Предоставление интерфейса для взаимодействия с аппаратным обеспечением компьютера
Символьные устройства	c	
Каналы	p	Организация взаимодействия процессов в операционной системе
Сокеты	s	

<http://younglinux.info>

Типы файлов в Linux

- **Обычные файлы** (регулярные) – любые текстовые, исполняемые, библиотечные, графические файлы;
- **Каталоги** – хранят именованные ссылки (только ссылки, но не сами файлы) на другие файлы;
- **Символьные ссылки** – файл с текстовой строкой, которая представляет собой путь к самому файлу;
- ...

Типы файлов в Linux

- **Сокеты** – файлы, которые используются для взаимодействия между различными процессами;
- **Именованные каналы FIFO** – подобны сокетам, но работают в одном направлении;
- **Файлы блочных и символьных устройств** – используются для взаимодействия с внешними устройствами.

Как создать файлы разных типов в Linux

	Тип файла	Команда
1	Regular file	touch file1
2	Directory	mkdir mydir1
3	Symbol link	ln -s file1 link_to_file1
4	Pipe	mkfifo mypipe
5	Socket	-
6	Block/symbol special file	mknod <name> c/b <major> <minor>

Как создать жесткую ссылку:

7	Hard link	ln file1 second_name_file1
---	-----------	----------------------------

Символические и жесткие ссылки

Жесткие ссылки	Символические ссылки
Не является отдельным типом файла	Является отдельным типом файла
Существует в рамках одной ФС	Может ссылаться на другие разделы и ФС
Представляет собой второе имя файла	Представляет собой ссылку или ярлык на файл
Не может указывать на каталог	Может указывать на каталог
При удалении уменьшается счетчик жестких ссылок(если 0 - то сам файл)	При удалении - удаляется файл-ярлык

Inode

Inode (index-node, индекс-узел, индексный дескриптор) — структура данных в ФС, в которой хранится метаданная о файлах каталогах и т.д.

Пример информации в inodes:

- размер файла;
- идентификатор (ID) устройства, содержащего файл;
- ID пользователя-владельца файл;
- указатели на блоки (кластеры) диска, в которых размещён файл;
- количество блоков, занимаемых файлом.

Inode

```
[art@localhost devops-basic]$ stat intro.md
  File: intro.md
  Size: 6917          Blocks: 16          IO Block: 4096   regular file
Device: 28h/40d Inode: 2961637      Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/   art)   Gid: ( 1000/   art)
Context: unconfined_u:object_r:user_home_t:s0
Access: 2021-11-26 10:51:03.673828798 +0300
Modify: 2021-10-22 01:01:56.985784376 +0300
Change: 2021-10-22 01:01:56.985784376 +0300
 Birth: 2021-10-22 01:01:56.985784376 +0300
```

Расшифровка:

- Context – контекст безопасности SELinux;
- Access – последний раз, когда к файлу обращались.
- Modify – последний раз, когда содержимое файла было изменено.
- Change – последний раз, когда атрибут файла или содержимое были изменены.
- Birth – Время создания файла (не поддерживается в Linux).

Команды Linux для работы с inodes

- **stat** – утилита позволяет просматривать состояние файла и даже файловой системы;
- **df -i** – утилита выводит информацию о файловых системах, их размере, занятом и свободном пространстве и точках монтирования;
- **ls -li** – утилита отображает список файлов в каталоге;
- **file -s /dev/sda3** - отобразить информацию о разделе



Итоги

Итоги

Сегодня мы познакомились с файловыми системами:

- для чего нужны ФС;
- какими свойствами они обладают;
- какие инструменты можно использовать для работы с ФС.