Файловая система Linux и структура каталогов

Про файлы

В файловой системе Linux существуют следующие типы файлов:

- обычные файлы (текстовые, картинки и т.п.)
- каталоги
- блочные устройства (представляют собой "драйверы" устройств. Блочное устройство производит чтение\запись в устройство блоками. Пример: жесткие диски, дискеты)
- символьные устройства (представляют собой "драйверы" устройств. Пример: терминалы, принтеры)
- символические ссылки
- PIPE (FIFO)
- гнезда (socket)

Тип файла в каталоге можно посмотреть командой ls с параметром -l.

Еще немного определений

Linux управляет всеми объектами в файловой системе через объект, называемый ***inode*** (сокращение от *index node*, **индексный дескриптор или индексный узел**).

Inode может ссылаться на файл, каталог или символическую ссылку на другой объект. Поскольку файлы используются для представления других типов объектов, например, устройств или памяти, inod'ы используются и для их представления.

Еще немного определений

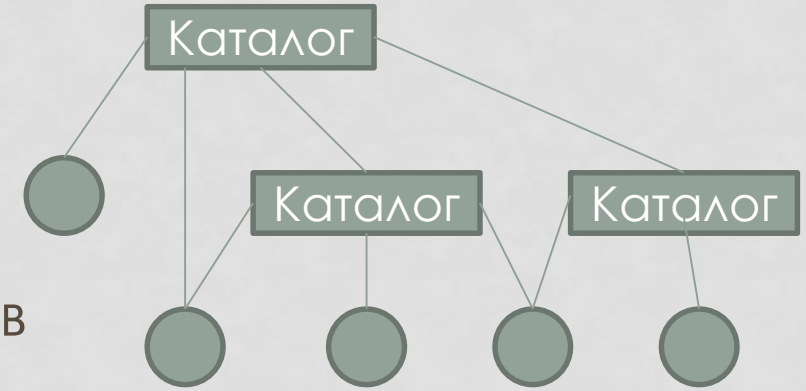
inode уникален в пределах определенной файловой системы.

Содержит следующую информацию:

- о владельце объекта ФС
- последнем времени доступа размере объекта ФС
- указании файл это или каталог
- права доступа

При перемещении файла (утилитой mv, например) в пределах одной файловой системы, inode файла остается неизменным (меняется только поле, описывающее имя путь файла), при перемещении файла в другую файловую систему сначала создается новый inode, а затем удаляется исходный.

Структура каталогов



Данная схема отображает то, что у одного объекта файловой системы может быть несколько путей. Т.е несколько файлов в структуре каталогов Linux могут быть физически одним файлом на диске.

Или другими словами, один физический файл на диске может иметь несколько имен (путей).

Это достигается тем, что в файловой системе каждый файл идентифицируется уникальным inode

Структура каталогов

Посмотреть сколько файл имеет ссылок и inode файла можно командой:

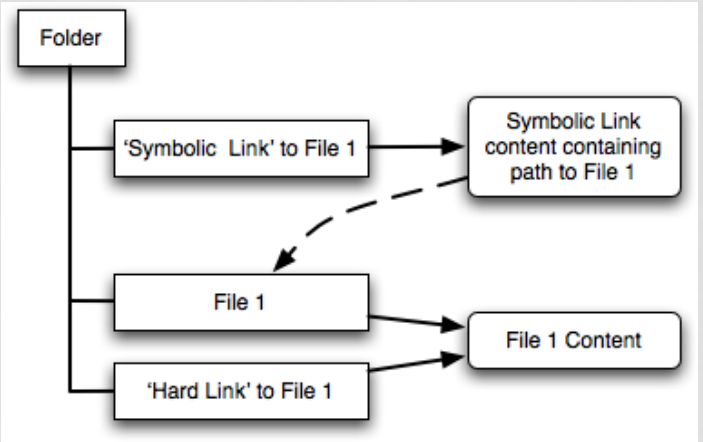
```
[antonk@home /]$ ls -li
193 drwxr-xr-x   1 antonk   root   368 Mar 30  2008 bin
    1 drwxr-xr-x   1 antonk   root     0 Jan   1  1970 dev
197 lrwxrwxrwx  1 antonk   root     7 Mar 30  2008 etc -> tmp/etc
```

в приведенном примере первый столбец (значения 193, 1, 197) – inode, а третий столбец (значения 1) – количество ссылок на файл (т.е. путей файла).

Лирическое отступление про ссылки

В Linux существует 2 вида ссылок:

- **Жесткая ссылка** (она же *Хардлинк, Hard-Link*) - это и есть один их путей файла (который указывается в команде ls -li)
- **Символьная** (она же *Симлинк* от *Symbolic link, символическая ссылка*) - это файл UNIX, содержащий в себе лишь текстовую строку - путь к оригинальному файлу, на который собственно ссылается.



Для создания ссылок воспользуйтесь командой ln

Жизнь файла

Файл в Linux существует пока на inode существует хотя бы одно указание (путь/имя).

Как только из системы удаляется последнее указание на inode, блоки, занимаемые файлом с данным inode "переходят" в свободный список (список блоков, доступных для выделения на диске). То есть блоки становятся свободным местом на диске.

Освободившийся inode тоже помещается в специальный список.

Структура каталогов

В файловой структуре Linux имеется один корневой раздел - / (он же корень). Все разделы жесткого диска (если их несколько) представляют собой структуру подкаталогов, "примонтированых" к определенным каталогам, схематично это можно представить следующим образом:

```
/-
|-/etc-|-/etc/X11-|-/etc/X11/xinit.d
|       |          |-...
|-/home <- |-/user1          # примонтированный раздел ext3,
|          |...               # содержащий свое дерево каталогов
|          |                  # (/home - точка монтирования)
|          |-/user2
|
|-....
```

Монтирование (mount)

Операция монтирования служит для того, чтобы сделать доступной файловую систему, расположенную на каком-либо блочном устройстве.

Суть операции монтирования заключается в том, что ядро ассоциирует некоторый каталог (называемый точкой монтирования) с блочным устройством и драйвером файловой системы. Ядро заносит в специальную таблицу монтирования информацию о том, что все файлы и каталоги, чей полный путь начинается с указанной точки монтирования, обслуживаются соответствующим драйвером файловой системы и расположены на указанном блочном устройстве.

Посмотреть таблицу примонтированных файловых систем можно через файл /proc/mounts.

fstab

При загрузке, ядро ОС Linux после монтирования корневого раздела на чтение, автоматически примонтирует остальные разделы жесткого диска. Описание ядро берет из файла /etc/fstab.

```
root@home:~# cat /etc/fstab
# /etc/fstab: sttic file system information.
proc            /proc           proc    defaults        0       0
/dev/hda6       /               reiserfs defaults       0       1
/dev/hda2       /boot           ext3    defaults        0       2
/dev/hda8       /dos            vfat    defaults        0       0
/dev/hda7       /home           xfs     defaults        0       2
/dev/hda1       /media/hda1     ntfs    defaults        0       0
/dev/hda5       none            swap    sw              0       0
/dev/hdc        /media/cdrom0   udf,iso9660 user,noauto     0       0
/dev/fd0        /media/floppy0  auto    rw,user,noauto  0       0
```

fstab (современный вариант)

```
[antonk@proxy  /]$ cat /etc/fstab
proc        /proc               proc    nosuid,noexec,gid=proc        0 0
devpts      /dev/pts            devpts  nosuid,noexec,gid=tty,mode=620  0 0
Tmpfs       /tmp                tmpfs   nosuid                        0 0
UUID=334fcb12-840a-474a-b278-089e86e6ad80    /   ext3    relatime     1       1
UUID=9bece036-5ecf-4440-ab51-00cd6c91b0b5   /home   ext3
nosuid,relatime,usrquota,grpquota       1       2
UUID=970184c4-6e91-4e6f-9681-bc1592dbb742     /var  ext3  nosuid,relatime  1   2
UUID=51318836-8b03-40e6-95a2-80498ab6e5db   /var/cache/squid       ext3
nodev,noexec,nosuid,relatime,usrquota,grpquota  1       2
UUID=1d50cadf-78ac-478a-a264-85dd64fd947b   swap    swap    defaults    0     0
/dev/sr0       /media/cdrom    udf,iso9660     ro,noauto,user,utf8     0    0
```

В современном варианте, устройства монтируются не по адресу устройства, а по UUID (идентификатору устройства). Это позволяет в случае смены жесткого диска, не перенастраивать fstab, а просто присвоить новому диску имеющийся UUID.

fstab

Строки содержат шесть полей. Все они должны быть заполнены.

file system - Монтируемое блочное устройство (файловая система). Можно вместо этого указывать идентификатор UUID. Это делает систему более устойчивой при установке и удалении устройств.

mount point - Это точка монтирования. Это путь, куда будет примонтирована файловая система **file system.** Для пространства подкачки это поле имеет значение none.

type - Определяет тип файловой системы. CD/DVD-диски часто имеют разные файловые системы - ISO9660 или UDF - поэтому вы можете перечислить различные возможности в виде списка, разделенного запятыми. Если вы хотите, чтобы mount автоматически определила тип, используйте auto, как сделано в последней строке для дискеты.

Option - Определяет параметры монтирования.

Dump - Определяет, будет ли команда dump включать данную файловую систему ext2 или ext3 в резервные копии. Значение 0 - игнорировать.

Pass - Ненулевые значения pass определяют порядок проверки файловых систем во время загрузки

fstab (options)

Опции монтирования – это специальные параметры, которые влияют на работу драйвера файловой системы. Для монтирования со значениями по умолчанию используйте defaults.

Несколько полезных опций:

- rw и ro указывают монтирование файловой системы в режиме чтения/записи или только для чтения.
- noauto указывает, что файловая система не должна автоматически монтироваться при загрузке или при выдаче команды mount -a. В нашем примере эта опция применена для съемных устройств.
- user определяет, что пользователь, не имеющий прав root, может монтировать данную файловую систему и демонтировать может только тот, кто смонтировал. Это особенно полезно для съемных носителей.
- users определяет, что любой пользователь, не имеющий прав root, может монтировать данную файловую систему и демонтировать. Это особенно полезно для съемных носителей.

UUID

UUID = Universally Unique Identifier

Используется чтобы описывать разделы в вашей ОС

Получить UUID разделов:

```
sudo blkid
/dev/sdb1: UUID="aabe7e48-2d11-421f-8609-7ea9d75e7f9b" TYPE="swap"
/dev/sdc1: UUID="9467f4de-4231-401f-bcaa-fee718d49e85" TYPE="ext4"
/dev/sdc3: UUID="93a54a4a-e0f5-4152-ae59-2245e8d16ee4" TYPE="ext4"
/dev/sde5: UUID="9467f4de-4231-401f-bcaa-fee718d49e85" TYPE="ext4"
/dev/sde6: LABEL="var" UUID="30433f28-1b79-4b4d-9985-fef5b1c886b5" TYPE="ext4"
```

Поменять UUID:

Сгенерировать новый:

```
uuidgen
f0acce91-a416-474c-8a8c-43f3ed3768f9
```

Изменить UUID раздела:

```
sudo tune2fs /dev/sde5 -U f0acce91-a416-474c-8a8c-43f3ed3768f9
```

Утилита mount

mount — специальная утилита которая используется для монтирования файловых систем.

Может использоваться отдельно от fstab

```
root@debian ~ # mkdir /ssd
root@debian ~ # mount /dev/sdc1 /ssd
root@debian ~ # mount | column -t
/dev/md1   on  /          type  ext4    (rw,errors=remount-ro)
tmpfs      on  /lib/init/rw  type  tmpfs   (rw,nosuid,mode=0755)
proc       on  /proc        type  proc    (rw,noexec,nosuid,nodev)
sysfs      on  /sys         type  sysfs   (rw,noexec,nosuid,nodev)
udev       on  /dev         type  tmpfs   (rw,mode=0755)
tmpfs      on  /dev/shm     type  tmpfs   (rw,nosuid,nodev)
devpts     on  /dev/pts     type  devpts  (rw,noexec,nosuid,gid=5,mode=620)
/dev/md0   on  /boot        type  ext2    (rw)
/dev/sdc1  on  /ssd         type  ext4    (rw)
```

Утилита mount

Можно настраивать атрибуты для монтируемых ФС с помощью опций монтирования. **mount** сама определяет файловую систему, но иногда это у нее не получается.

Опция -t позволяет задать тип файловой системы.
Опция -o определяет атрибуты доступа в фс.
Опция -a монтирует все фс указанных в файле /etc/fstab.

```
root@debian ~ # mount -t ext4 -o rw,noexec,nosuid /dev/sdc1 /ssd
root@debian ~ # mount | column -t
/dev/md1 on / type ext4 (rw,errors=remount-ro)
tmpfs on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
udev on /dev type tmpfs (rw,mode=0755)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=620)
/dev/md0 on /boot type ext2 (rw)
/dev/sdc1 on /ssd type ext4 (rw,noexec,nosuid)
```

Разбить диск и форматировать

Для того что бы размещать файлы на жестком диске или другом носителе например флешке нам нужно как минимум проделать несколько этапов а именно – разбить диск на разделы (**создать таблицу разделов**) и создать файловую систему (**форматирование**) с последующим монтированием в систему.

Обзор блочных устройств

Посмотреть какие есть блочные устройства в системе можно так:

```
root@debian ~ # ls -l /dev/sd*
brw-rw---- 1 root disk 8,  0 Map 11 19:03 /dev/sda
brw-rw---- 1 root disk 8,  1 Map 10 21:23 /dev/sda1
brw-rw---- 1 root disk 8,  5 Map 10 21:23 /dev/sda5
brw-rw---- 1 root disk 8, 16 Map 10 22:31 /dev/sdb
brw-rw---- 1 root disk 8, 17 Map 10 21:23 /dev/sdb1
brw-rw---- 1 root disk 8, 18 Map 10 21:23 /dev/sdb2
brw-rw---- 1 root disk 8, 32 Map 11 19:05 /dev/sdc
brw-rw---- 1 root disk 8, 33 Map 11 17:59 /dev/sdc1
```

b – блочное устройство, судя по выводу у нас 3 физических диска.

Или

```
fdisk –l
```

Вывестии информацию о жестком диске можно так

```
hdparm -I /dev/sdX
```

Разбить диск

Для этого можете использовать fdisk или cfdisk.

Пример:

```
root@debian ~ # fdisk /dev/sdc
Command (m for help): n
p primary partition (1-4)
Partition number (1-4): 1
First cylinder (1-7297, default 1):1
Last cylinder, +cylinders or +size{K,M,G} (1-7297, default
7297): +10G
Command (m for help): w
```

Мы создали раздел размером 10гб.

Проверить: `fdisk -l /dev/sdc`

Форматировать раздел

Форматировать и создавать новую ФС будем при помощи утилиты mkfs.

mkfs + tab

root@debian ~ # mkfs

mkfs mkfs.cramfs mkfs.ext3 mkfs.ext4dev

mkfs.bfs mkfs.ext2 mkfs.ext4 mkfs.minix

Форматировать раздел

```
root@debian ~ # mkfs.ext4 /dev/sdc1
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
3670016 inodes, 14653280 blocks
732664 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
448 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
4096000, 7962624, 11239424

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```