

Linux. Циклическое устройство

20.08.2023
Теги: CLI • Linux • БлочноеУстройство • Команда • Файл • ФайловаяСистема

Linux позволяет создать специальное блочное устройство, которое может отображать блоки данных обычного файла или другого блочного устройства. Это может быть полезно для получения образа файловой системы, хранящейся в файле, в виде блочного устройства. Это блочное устройство может быть смонтировано с помощью команды `mount` – для дальнейшей работы.

Первая часть

Давайте создадим файл, отобразим его блоки в loop-устройство, создадим разделы и файловые системы, запишем туда что-нибудь.

```
$ mkdir /home/evgeniy/data
$ cd /home/evgeniy/data
$ dd if=/dev/zero of=part-home-data bs=1M count=2000
```

Теперь у нас есть файл `part-home-data`, у которого 2000 блоков размером 1Мб, заполненный нулями. Давайте создадим loop-устройство, которое будет отображать блоки нашего файла. Опция `-f` позволяет найти первое неиспользуемое loop-устройство, чтобы использовать его для отображения файла.

```
$ sudo losetup -f part-home-data
```

Примечание

Опция `-P` заставит ядро сканировать таблицу разделов на вновь созданном loop-устройстве – это полезно, если файл `part-home-data` уже содержит разделы, которые были созданы ранее.

Опция `-a` позволяет вывести список используемых loop-устройств

```
$ losetup -a
/dev/loop0: []: (/home/evgeniy/data/part-home-data)
```

Теперь создаем таблицу разделов GPT и два раздела, каждый из которых будет занимать 50% места

```
$ sudo parted --align opt /dev/loop0
GNU Parted 3.4
```

Категории блога

- Web-разработка
 - HTML и CSS
 - JavaScript и TypeScript
 - PHP и MySQL
 - CMS Битрикс
 - CMS WoprdPress
 - Yii2 и Laravel
 - Python и Flask
 - Web-аналитика
 - Разное
- ОС Linux
 - Команды
 - Сценарии
 - Разное
- 1С:Предприятие
 - Программирование
 - Язык запросов
 - Разное
- Локальная сеть
 - Разное

Облако тегов

- | | |
|----------------------|-----------------------|
| 1С:Предприятие (31) | Yii2 (69) |
| API (29) | БазаДанных (95) |
| Bash (43) | Битрикс (66) |
| CLI (124) | Блог (29) |
| CMS (139) | Верстка (43) |
| CSS (50) | ИнтернетМагаз... (84) |
| Frontend (75) | КаталогТоваров (87) |
| HTML (66) | Класс (30) |
| JavaScript (150) | Клиент (28) |
| Laravel (72) | Ключ (28) |
| Linux (171) | Команда (88) |
| MySQL (76) | Компонент (60) |
| PHP (125) | Конфигурация (66) |
| React.js (66) | Корзина (32) |
| SSH (27) | ЛокальнаяСеть (32) |
| Ubuntu (69) | Модуль (34) |
| Web-разработка (509) | Навигация (31) |
| WordPress (73) | Настройка (143) |

```
Используется /dev/loop0
Добро пожаловать в GNU Parted! Наберите 'help' для просмотра с
```

```
(parted) mktable gpt
(parted) mkpart home 0% 50%
(parted) mkpart data 50% 100%
```

```
(parted) print
Модель: Устройство типа loopback (loopback)
Диск /dev/loop0: 2097MB
Размер сектора (логич./физич.): 512B/512B
Таблица разделов: gpt
Флаги диска:
```

Номер	Начало	Конец	Размер	Файловая система	Имя	Флаги
1	1049kB	1049MB	1048MB		home	
2	1049MB	2096MB	1048MB		data	

```
(parted) quit
```

ПанельУправле... (29) Установка (67)

Плагин (33) Файл (51)

Пользователь (26) Форма (58)

Практика (101) Фреймворк (192)

Сервер (77) Функция (36)

Событие (28) ШаблонСайта (68)

Теория (106)

Все теги

Категории статей

Web-разработка

1С:Предприятие

ОС Linux

Регулярные выражения

Разное

Теперь создаем в этих разделах файловые системы

```
$ sudo mkfs.ext4 /dev/loop0p1
mke2fs 1.46.5 (30-Dec-2021)
Discarding device blocks: done
Creating filesystem with 255744 4k blocks and 64000 inodes
Filesystem UUID: 42d3f81d-0367-47cf-aacb-acb39154fd27
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Сохранение таблицы inod'ов: done
Создание журнала (4096 блоков): готово
Writing superblocks and filesystem accounting information: rot
```

```
$ sudo mkfs.ext4 /dev/loop0p2
mke2fs 1.46.5 (30-Dec-2021)
Discarding device blocks: done
Creating filesystem with 255744 4k blocks and 64000 inodes
Filesystem UUID: d2e9cd92-b455-4f6a-9b17-32df86b8aced
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Сохранение таблицы inod'ов: done
Создание журнала (4096 блоков): готово
Writing superblocks and filesystem accounting information: rot
```

Теперь создаем две точки монтирования, чтобы смонтировать в них два раздела

```
$ sudo mkdir /mnt/home
$ sudo mkdir /mnt/data
```

Почти все готово, осталось только смонтировать и записать туда что-нибудь

```
$ sudo mount /dev/loop0p1 /mnt/home  
$ sudo mount /dev/loop0p2 /mnt/data
```

Копировать

```
$ sudo touch /mnt/home/homefile  
$ sudo touch /mnt/data/part-home-data
```

Копировать

Размонтируем файловые системы

```
$ sudo umount /mnt/home  
$ sudo umount /mnt/data
```

Копировать

Вторая часть

Создание разделов необязательно – можно было просто создать файловую систему

```
$ cd /home/evgeniy/data  
$ dd if=/dev/zero of=filesystem bs=1M count=1000  
$ sudo losetup -f filesystem  
$ losetup -a  
/dev/loop1: []: (/home/evgeniy/data/filesystem)  
/dev/loop0: []: (/home/evgeniy/data/part-home-data)  
$ sudo mkfs.ext4 /dev/loop1 # создаем файловую систему  
mke2fs 1.46.5 (30-Dec-2021)  
Discarding device blocks: done  
Creating filesystem with 256000 4k blocks and 64000 inodes  
Filesystem UUID: 4a2f80ea-1f2f-461c-aa00-8a69313af0c8  
Superblock backups stored on blocks:  
    32768, 98304, 163840, 229376  
  
Allocating group tables: done  
Сохранение таблицы inod'ов: done  
Создание журнала (4096 блоков): готово  
Writing superblocks and filesystem accounting information: rot  
$ sudo mkdir /mnt/filesystem # создаем точку монтирования  
$ sudo mount /dev/loop1 /mnt/filesystem # монтируем  
$ sudo touch /mnt/filesystem/somedata # создаем файл
```

Копировать

Размонтируем файловую систему

```
$ sudo umount /mnt/filesystem
```

Копировать

Третья часть

Освобождаем loop-устройства

```
$ sudo losetup -d /dev/loop0
```

Копировать

```
$ sudo losetup -d /dev/loop1
```

Копировать

Четвертая часть

Пример из второй части можно выполнить проще, если монтировать файл с файловой системой внутри, используя опцию монтирования `loop`.

```
$ sudo mount -o loop /home/evgeniy/data/filesystem /mnt/filesystem
```

```
$ ls -la /mnt/filesystem
итого 24
drwxr-xr-x 3 root root 4096 авг 20 12:12 .
drwxr-xr-x 7 root root 4096 авг 20 13:16 ..
drwx----- 2 root root 16384 авг 20 12:07 lost+found
-rw-r--r-- 1 root root 0 авг 20 12:12 somedata
```

С примером из первой части будет сложнее – команде `mount` нужно указать, где начинается файловая система и сколько места занимает.

```
$ sudo fdisk -l /home/evgeniy/data/part-home-data
Диск /home/evgeniy/data/part-home-data: 1,95 GiB, 2097152000 байт
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
Тип метки диска: gpt
Идентификатор диска: 14C53831-DFE1-42B6-9CE4-E73AC046433E

Устр-во          Начало    Конец Секторы Раз
/home/evgeniy/data/part-home-data1 2048 2047999 2045952 9
/home/evgeniy/data/part-home-data2 2048000 4093951 2045952 9
```

Первая файловая система начинается с блока 2048, длина файловой системы 2045952 блока, каждый блок 512 байт.

```
$ echo $((2048 * 512)) $((2045952 * 512))
1048576 1047527424
```

```
$ sudo mount -o loop,offset=1048576,sizelimit=1047527424 /home/evgeniy/data/part-home-data /mnt/home
```

```
$ ls -la /mnt/home
итого 24
drwxr-xr-x 3 root root 4096 авг 20 11:58 .
drwxr-xr-x 8 root root 4096 авг 23 16:36 ..
-rw-r--r-- 1 root root 0 авг 20 11:58 homefile
drwx----- 2 root root 16384 авг 20 11:45 lost+found
```

Вторая файловая система начинается с блока 2048000, длина файловой системы 2045952 блока, каждый блок 512 байт.

```
$ echo $((2048000 * 512)) $((2045952 * 512))
1048576000 1047527424
```

```
$ sudo mount -o loop,offset=1048576000,sizelimit=1047527424 /home/evgeniy/data/part-home-data2 /mnt/data
```

```
$ ls -la /mnt/data
итого 24
drwxr-xr-x 3 root root 4096 авг 20 11:58 .
drwxr-xr-x 8 root root 4096 авг 23 16:36 ..
```

```
-rw-r--r-- 1 root root    0 авг 20 11:58 datafile
drwx----- 2 root root 16384 авг 20 11:47 lost+found
```

Все работает, все файлы, которые были созданы в первой и второй части на месте, можно размонтировать.

```
$ sudo umount /mnt/filesystem
$ sudo umount /mnt/home
$ sudo umount /mnt/data
```

[Копировать](#)

Похожие записи

- Linux. Область подкачки
- Linux. Специальные файловые системы
- LVM, часть 3 из 3. Практика
- LVM, часть 2 из 3. Практика
- LVM, часть 1 из 3. Теория
- Linux. Команда mdadm
- Linux. Команда mount

Поиск: CLI • Linux • Команда • Файловая система • loop • losetup • Блочное устройство

Узелки на память: Web-разработка, 1С:Предприятие, ОС Linux