

# **ОПЕРАЦИОННЫЕ СИСТЕМЫ И СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ**

**Маленькие хитрости**

**Преподаватель: Поденок Леонид Петрович, 505а-5**

**+375 17 293 8039 (505а-5)**

**+375 17 320 7402 (ОИПИ НАНБ)**

**prep@lsi.bas-net.by**

**ftp://student:2ok\*uK2@Rwox@lsi.bas-net.by**

**Кафедра ЭВМ, 2024**

# Оглавление

Как использовать регулярный файл вместо блочного устройства.....	3
--	---

# Как использовать регулярный файл вместо блочного устройства

Системные вызовы `open()`, `read()`, `write()`, `lseek()` почти одинаково работают как на регулярных файлах, так и на блочных устройствах. Когда утилита форматирования устройства `mkfs(8)` создает файловую систему на устройстве, она пользуется именно этими вызовами. Иными словами, блочное устройство не отличается в отношении чтения/записи от регулярного файла.

Чтобы использовать файл в качестве блочного устройства, можно поступить следующим образом:

1) создаем файл требуемого размера, например, 1G, и форматируем его, например, в FAT16:

```
$ head -c 1G /dev/zero > file
$ ls -l file
-rw-r--r--. 1 user user 1073741824 apr 10 15:18 file
$ file file
file: data
$ mkfs.fat -F 16 -n FILE-FAT16 file
$ file file
file: DOS/MBR boot sector, code offset 0x3c+2, OEM-ID "mkfs.fat", sectors/cluster
32, reserved sectors 32, root entries 512, Media descriptor 0xf8, sectors/FAT
256, sectors/track 63, heads 64, sectors 2097144 (volumes > 32 MB), serial number
0x5ad961a9, label: "FILE-FAT16 ", FAT (16 bit)
```

Можно совместить обе операции и сразу создать и файл и его отформатировать:

```
$ mkfs.fat -C -F 16 -n FILE-FAT16 file2 1048576
```

В результате будет создан файл и файловая система. Оба варианта производят совершенно одинаковый (с точностью до серийного номера) выхлоп.

2) Теперь созданную в файле файловую систему можно смонтировать, например, в точку монтирования /tmp/file-fat16:

```
$ mkdir /tmp/file-fat16
$ sudo mount file /tmp/file-fat16
```

Посмотрим, как была смонтирована созданная в файле файловая система и какие права на нее устанавливаются – нам чуть позже понадобится ее UUID:

```
$ lsblk -f
NAME FSTYPE FSVER LABEL          UUID                                FSAVAIL FSUSE% MOUNTPOINTS
...
loop1 vfat    FAT16 FILE-FAT16 5AD9-61A9                        1023,7M    0% /tmp/file-fat16
...
$ ls -ld /tmp/file-fat16
drwxr-xr-x. 2 root root 16384 янв  1 1970 /tmp/file-fat16
```

Странная дата, однако именно так в Микрософт задумано для dos-утилиты format, поведение которой эмулируется утилитой mkfs.fat. Неприятно, но так уж есть.

Поскольку монтирование было выполнено от имени привилегированного пользователя, все действия в смонтированной файловой системе нужно выполнять от его имени. Чтобы созданную файловую систему можно было монтировать от имени простого пользователя, привилегированному пользователю потребуется проделать некоторые дополнительные действия.

3) Сначала определим, куда выполняется автомонтирование съемных устройств, куда наш пользователь будет монтировать файл с файловой системой. В Fedora Linux каталог автомонтирования для пользователя user расположен в /run/media/user. Создаем там каталог для монтирования нашего файла и разрешим пользователю полный доступ к нему:

```
$ sudo mkdir -p /run/media/user/file-fat16
$ sudo chown user:user /run/media/user/file-fat16
```

4) Чтобы пользователь мог монтировать файл простой командой

```
$ mount file
```

необходимо в файл /etc/fstab добавить строку и перечитать /etc/fstab:

```
UUID=5AD9-61A9 /run/media/user/file-fat16 vfat rw,suid,user,exec,noauto,async,owner 0 2
$ sudo systemctl daemon-reload
```

5) Теперь можно монтировать и простому пользователю

```
$ mount file
$ lsblk -f
NAME  FSTYPE FSVER LABEL      UUID          FSAVAIL FSUSE% MOUNTPOINTS
...
loop0 vfat    FAT16 FILE-FAT16 5AD9-61A9 1023,7M      0% /run/media/user/file-fat16
      /tmp/file-fat16
...
```

В результате имеем смонтированную ФС в две точки монтирования одного устройства loop0.

Чтобы удалить любую из них или обе, требуется выполнение двух запусков `umount`. Поскольку монтирование в `/tmp/file-fat16` было выполнено от имени суперпользователя, размонтирование нужно выполнять именно от него:

```
$ sudo umount /dev/loop0    # размонтирует ФС в /run/media/user/file-fat16
$ sudo umount /dev/loop0    # размонтирует ФС в /tmp/file-fat16
```

Размонтирование происходит в порядке, обратном монтированию. Если нам нужно размонтировать конкретную точку монтирования, мы указываем ее при запуске `umount`:

```
$ sudo umount /tmp/file-fat16
```

Смонтированное пользователем может этим пользователем и размонтироваться. Смонтированное пользователем также можно размонтировать из окна управления устройствами или кликнув на иконку на рабочем столе. В частности, в `xfce4` для этой цели можно использовать `xfce4-mount-plugin`.