

chuzhoy007.ru /komanda-mkfs-linux

## Команда mkfs Linux для форматирования разделов и дисков



Mkfs — это утилита командной строки в UNIX и Unix-подобных операционных системах. Например Linux, FreeBSD и т.п.. Используется для создания файловой системы на блочном устройстве, таком как жесткий диск или флэш-накопитель. Mkfs — это сокращение от «make file system» (создание файловой системы).

### Для чего применяется mkfs

Mkfs необходим для создания файловой системы на диске перед его использованием. По другому можно сказать — форматирование (создание структуры). Без создания файловой системы, устройство не может быть использовано для хранения данных.

Файловая система — это организация данных на устройстве хранения, таком как жесткий диск или флэш-накопитель. Она определяет, как информация будет храниться и организовываться на устройстве, а также указывает права для доступа чтения и записи.

Утилита позволяет выбрать тип файловой системы, которую нужно создать. Это может быть необходимо, чтобы оптимизировать работу устройства, совместимость с другими системами или для обеспечения безопасности хранения данных.

## Синтаксис

Синтаксис команды mkfs выглядит следующим образом:

```
1 mkfs [опции] [-t тип_фс] [устройство] [размер блока]
```

Здесь:

- **опции** – дополнительные опции команды;
- **устройство** – устройство, на котором нужно создать файловую систему;
- **размер блока** – размер блока в байтах. Этот параметр может быть опущен, в этом случае будет использован размер блока по умолчанию. Размер блока зависит от файловой системы. В большинстве случаев это 4 КБ (4096 байт).

разберем основные опции:

- **-t <тип>**: задает тип файловой системы, которую необходимо создать. Например, ext4, ntfs, fat32, xfs, btrfs и т.д. Если этот параметр не указан, команда сама определит тип файловой системы на основе типа устройства;
- **-n <имя>**: задает метку (имя) файловой системы.
- **-b <размер>**: задает размер блока файловой системы в байтах
- **-c**: проверяет блоки на чтение при создании файловой системы.
- **-i <число>**: задает интервал между резервными копиями суперблоков. Эта опция используется для предотвращения повреждения файловой системы.
- **-L <метка>**: задает метку файловой системы (аналогично опции -n).
- **-v**: выводит подробную информацию о процессе создания файловой системы.
- **-f**: позволяет выполнить быстрое форматирование без проверки на ошибки.

Некоторые опции могут быть несовместимы между собой. Например, опции -n и -v не могут быть использованы вместе, поскольку опция -n отменяет создание файловой системы, тогда как опция -v выводит подробную информацию о создаваемой файловой системе.

Команда mkfs также может использоваться для создания файловой системы на RAID-массиве. В этом случае в качестве пути к блочному устройству указывается путь к RAID-массиву, а не к физическому диску.

## Основные файловые системы поддерживаемые mkfs

Mkfs поддерживает большое количество файловых систем. Каждая имеет свои преимущества и недостатки поэтому используйте ту или другую в зависимости от задач для которых будет применяться устройства. Рассмотрим основные современные файловые системы применяемые для работы операционных систем:

- **Ext2** (Second Extended File System) — это стандартная файловая система Linux. Она имеет ограниченную поддержку журналирования, что может привести к потере данных при сбоях. Однако, она предоставляет высокую производительность при работе с маленькими

файлами.

- **Ext3** (Third Extended File System) — это улучшенная версия Ext2 с поддержкой журналирования. Это обеспечивает повышенную надежность, потому что журнал позволяет восстановить файловую систему в случае сбоев.
- **Ext4** (Fourth Extended File System) — это еще более усовершенствованная версия Ext3 с улучшенной производительностью и поддержкой больших файлов. Ext4 поддерживает журналирование и другие функции, такие как разделение блоков и увеличение максимального размера файловой системы.
- **XFS** (Extended File System) — это файловая система, разработанная для работы с большими файлами и высокой производительности. Она используется в крупных корпоративных средах для обработки больших объемов данных, например, в хранилищах данных или в системах потоковой передачи медиа-файлов.
- **Btrfs** (B-tree file system) — это новая файловая система, которая предлагает современные возможности, такие как управление снимками и копирование по записи. Эта файловая система может использоваться для хранения больших объемов данных, в том числе в облаке.
- **NTFS** (New Technology File System) — это файловая система, разработанная компанией Microsoft. NTFS используется для хранения данных в Windows и поддерживает различные функции, такие как шифрование и компрессия.

Читайте также: [Как установить Viber на компьютер в Debian linux](#)

Кроме того, mkfs поддерживает файловые системы FAT16, FAT32, exFAT и другие, которые используются на флэш-накопителях и портативных устройствах.

Файловая система FAT16 (File Allocation Table) была разработана Microsoft для использования на дисководах и дискетах. Она поддерживает ограниченный размер файлов и дисков, поэтому ее редко используют для современных устройств хранения данных.

Файловая система FAT32 была создана как улучшенная версия FAT16. Она поддерживает большие файлы и диски, и широко используется на флэш-накопителях.

Файловая система exFAT (Extended File Allocation Table) была разработана Microsoft для использования на флэш-накопителях и других устройствах с большими файлами. exFAT поддерживает файлы размером более 4 Гб, и при этом обеспечивает совместимость между различными операционными системами.

Таким образом, mkfs предоставляет широкие возможности для создания различных файловых систем в Linux, что позволяет оптимизировать работу с различными вариантами устройств хранения данных в соответствии с их особенностями.

## Примеры использования

**ВНИМАНИЕ!** При форматировании разделов вся информация будет удалена! Перед использованием утилиты сделайте резервные копии важных файлов и убедитесь что вся информация скопирована в надежный источник и никаких нужных файлов нет на

устройстве с которым вы будете работать!

Рассмотрим практическое применение команды `mkfs` для работы с дисковыми системами компьютера. Первое, что необходимо сделать это определить название диска. Сделать это можно с помощью команды `lsblk`.

```
Терминал - user@deb: ~
Файл Правка Вид Терминал Вкладки Справка
user@deb:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   25G  0 disk
├─sda1       8:1    0   24G  0 part /
├─sda2       8:2    0    1K  0 part
└─sda5       8:5    0  975M  0 part [SWAP]
sdb          8:16   0   30G  0 disk
sr0         11:0    1 1024M  0 rom
user@deb:~$
```

<http://chuzhoy007.ru>

Либо с помощью команды:

```
1 $ sudo fdisk -l
```

Я уже писал про неё. Читайте подробный [обзор утилиты fdisk](#).

```
Терминал - user@deb: ~
Файл Правка Вид Терминал Вкладки Справка
user@deb:~$ sudo fdisk -l
[sudo] пароль для user:

Disk /dev/sda: 25 GiB, 26843545600 bytes, 52428800 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xfc01d0ae

Device      Boot      Start         End      Sectors   Size Id Type
/dev/sda1   *          2048    50427903   50425856    24G 83 Linux
/dev/sda2             50429950   52426751    1996802   975M  5 Extended
/dev/sda5             50429952   52426751    1996800    975M 82 Linux swap / Solaris

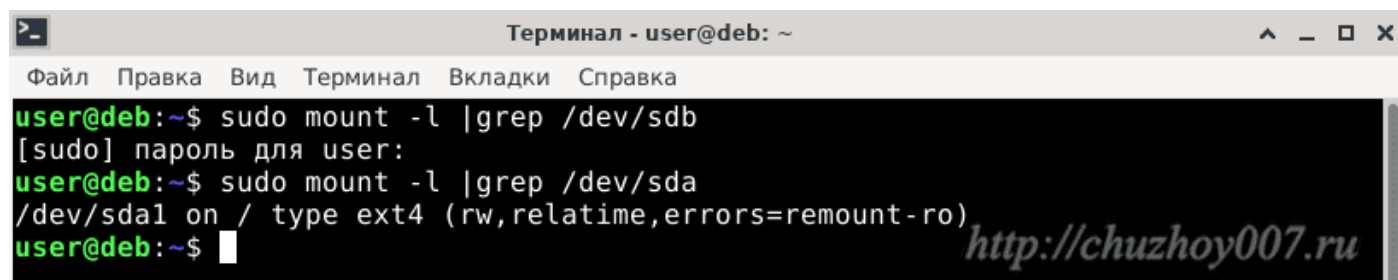
Disk /dev/sdb: 30 GiB, 32212254720 bytes, 62914560 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
user@deb:~$
```

<http://chuzhoy007.ru>

Работать с дисками можно только если они не смонтированы в системе. Я покажу как проверить является ли диск примонтированным или нет на примере диска `/dev/sdb`. Команда `mount`:

```
1 $ sudo mount -l | grep /dev/sdb
```

Если диск не примонтирован вывод будет пустой. Если примонтирован то будут отображены параметры монтирования. Смотрите скриншот ниже. Для наглядности на нем приведено отображение двух дисков. Первая строка /dev/sdb не смонтирован и вторая строка /dev/sda смонтирован.



```
Терминал - user@deb: ~
Файл Правка Вид Терминал Вкладки Справка
user@deb:~$ sudo mount -l |grep /dev/sdb
[sudo] пароль для user:
user@deb:~$ sudo mount -l |grep /dev/sda
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro)
user@deb:~$
```

Чтобы размонтировать диск используйте команду:

```
1 $ sudo umount /dev/sdb
```

вместо /dev/sdb укажите путь до вашего устройства.

Теперь можно форматировать диски и разделы.

Читайте также: [Команда fdisk в linux для работы с дисками](#)

Для примера я создам два раздела на диске /dev/sdb. Отформатирую один один раздел в файловую систему ext4 для хранения данных. Второй — в файловую систему btrfs и задам метку для диска bckp для создания бэкапов.

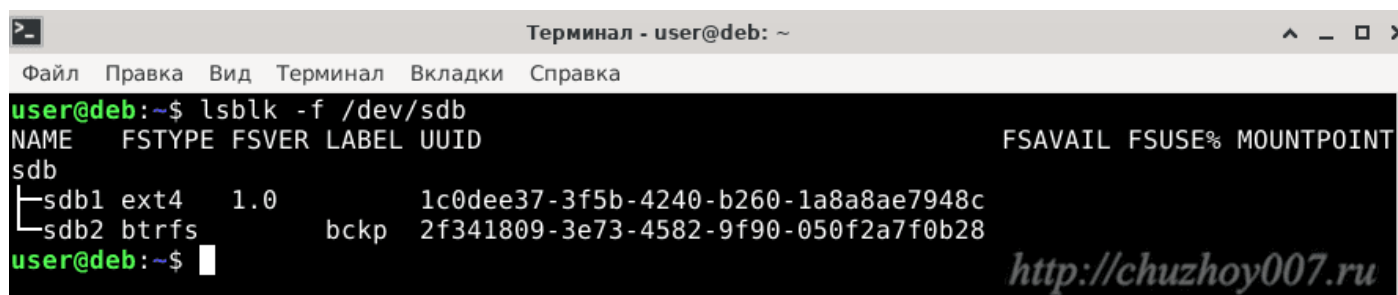
Первая команда:

```
1 $ sudo mkfs -t ext4 /dev/sdb1
```

Вторая команда:

```
1 $ sudo mkfs -t btrfs -L bckp /dev/sdb2
```

С помощью команды lsblk -f /dev/sdb посмотрим на результат.



```
Терминал - user@deb: ~
Файл Правка Вид Терминал Вкладки Справка
user@deb:~$ lsblk -f /dev/sdb
NAME      FSTYPE FSVER LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINT
sdb
├─sdb1    ext4    1.0          1c0dee37-3f5b-4240-b260-1a8a8ae7948c
└─sdb2    btrfs          bckp  2f341809-3e73-4582-9f90-050f2a7f0b28
user@deb:~$
```

Как видно на скриншоте все получилось как и было запланировано.

## Дополнительные короткие команды

Помимо вышеописанной универсальной формы для форматирования есть и специально

созданные узконаправленные команды для создания самых распространенных файловых систем:

- **mkfs.ext2**, **mkfs.ext3**, **mkfs.ext4** — команды для форматирования разделов в файловые системы ext2, ext3 и ext4 соответственно;
- **mkfs.fat**, **mkfs.vfat**, **mkfs.msdos** — команды для форматирования разделов в файловые системы FAT и VFAT;
- **mkfs.ntfs** — команда для форматирования разделов в файловую систему NTFS;
- **mkfs.xfs** — команда для форматирования разделов в файловую систему XFS;
- **mkswap** — команда для форматирования разделов под swp-раздел (раздел подкачки).

Например, чтобы быстро создать раздел ext4, на блочном устройстве, с параметрами по умолчанию можно дать команду:

```
1 $ sudo mkfs.ext4 /dev/sdb
```

На диске будет создан раздел с файловой системой ext4. И так по аналогии можно создавать другие.