

# **ASTRA LINUX**

# **GUIDE**

**DISK**

## Просмотр свободного места на диске/разделах в Linux

В данной статье мы расскажем, как проверить свободное дисковое пространство на хосте под управлением Linux через консоль. Мы рассмотрим несколько базовых команд с помощью которых можно узнать размер не только одного диска или директории, а целого дерева каталогов. Отдельно выделим комбинированные команды для получения более детальной информации о размерах вложенных папок. В конце статьи мы приведем пример простого скрипта для мониторинга дискового пространства с отправкой уведомлением на почтовый ящик администратора.

Просмотр занятого места на разделах командой du

Выведем список дисков в системе и их размер:

```
# fdisk -l
```

Чтобы получить более детальную информацию по двум разделам, которые созданы на диске, можно использовать:

```
# lsblk
```

```
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda 253:0 0 40G 0 disk
  vda1 253:1 0 512M 0 part /boot
  vda2 253:2 0 39.5G 0 part /
```

В листинге команды видно, что один раздел используется под /boot, а второй под корень системы.

Перейдем к утилите, вокруг которой мы построили данную инструкцию:

df – это команда позволяет отобразить информацию о свободном/доступном месте на диске, файловой системе раздела.

Чтобы листинг команды был более читабелен, нужно использовать ее с опциями.

Например:

```
# df -h
```

Описание столбцов:

- Filesystems – имя файловой системы
- Size – размер раздела
- Used – используемое дисковое пространство
- Avail – доступное дисковое пространство(свободное)
- Use% — занятое дисковое пространство в процентах
- Mounted on – смонтировано. Указывает директорию, к которой примонтирован раздел.

При использовании опции -h дисковое пространство выводится в ГБ. Если размер меньше 1Гб, то вывод будет в Мб.

Дополнительные опции:

- df -m – информация будет отображена в Мб.
- df -k — информация будет отображена в Кб.
- df -T – к выводу добавиться тип файловой системы
- df -hT /boot – с данной опцией, будет предоставлена информация

Опции можно комбинировать между собой, для получения нужной информации.

Полная справка по df доступна через команду:

```
# df --help
```

Получаем размеры директорий командой du

Вторая команда, которую мы рассмотрим, это du. du – команда отображающая размер директории.

Дополнительные опции для du:

- du -h – показывает информацию в читабельном формате. Можно указать путь к конкретной директории, если применить без этого, покажет информацию в текущем каталоге.
- du -sh – покажет размер директории в которой вы находитесь. Можно указать путь к директории.
- du -m – вывод информации в Мб.
- du -h --time – выведет информацию с дополнительным столбцом по времени изменения директории или файлов.

При администрировании сервера, можно комбинировать вышеописанные команды.

Например, выведем размер всех вложенных каталогов в /var/log и отсортируем вывод от большего к меньшему (чтобы самые большие каталоги оказались вверху списка):

```
# du -h /var/ | sort -rn
```

```
# du --max-depth=1 -h – выводит на экране размер вложенных директории, не дальше 1-го уровня вложенности.
```

Таким образом, вы легко можете определить сколько места занимают разные директории, и быстрой найти большие каталоги, которые занимают очень много места.

Bash скрипт мониторинга свободного места в Linux

Рассмотрим простой bash скрипт, для мониторинга дискового пространства на сервере и отправке email уведомления администратору, если места недостаточно.

```
#!/bin/bash
CURRENT=$(df / | grep / | awk '{ print $5}' | sed 's/%//g')
THRESHOLD=90
if [ "$CURRENT" -gt "$THRESHOLD" ] ; then
mail -s 'Заканчивается дисковое пространство' -r admin@example.com << EOF
В вашем корневом разделе сервера server1, осталось слишком мало дискового
пространства. Используется: $CURRENT%
EOF
fi
```

При достижении заполняемости директории / до 90% к вам на почту будет отправляться уведомление. Добавьте данный скрипт в cron и мониторьте вашу систему. Можно изменить скрипт, чтобы он смотрел за размером определенного раздела или каталога.

## Проверка диска на ошибки

1. Введите следующую команду, чтобы узнать список дисков, найти нужный диск/раздел и определить его файловую систему:

```
fdisk -l | more
```

2. Если хотите проверить, правильно ли выбрали диск по содержимому, выполните (например, касательно диска /dev/sda2):

```
# mount /dev/sda2 /mnt  
# ls /mnt  
# umount /mnt
```

3. В зависимости от файловой системы (см. результат команды fdisk из пункта 1) примените к диску следующую команду:

### 3.1 NTFS-диск

```
ntfsresize -i -f -v /dev/sda1
```

Хотя желательней выполнить проверку из-под Windows NT

```
chkdsk c: /f /r
```

### 3.2 ext2/3/4

```
e2fsck -f -y -v -C 0 /dev/sda7
```

-f — принудительная проверка файловой системы, даже если по журналу она чистая;  
-у — отвечает на все вопросы об исправлении ошибок утвердительно, можно сказать, что это эквивалент а.\*

-v — максимально подробный вывод.

-C — показать прогресс проверки файловой системы

### 3.3.FAT\*

```
fsck.fat -a -w -v /dev/sdc1
```

С помощью Shift+PgUp/PgDn можно пролистывать консоль вверх/вниз.

## МОНТИРОВАНИЕ ДИСКА В LINUX

Разделы дисков в Linux подключаются к системе совсем не так, как в Windows. Здесь есть корневая файловая система, куда подключаются все другие разделы и устройства, которые вы будете использовать. Системные разделы монтируются автоматически при старте системы. Но если вам нужно подключить дополнительные разделы, в некоторых случаях, может понадобиться это делать вручную.

### ЧТО ТАКОЕ МОНТИРОВАНИЕ?

Как я уже сказал Linux имеет единую корневую файловую систему, куда подключаются все запоминающие устройства и другие ресурсы. На самом деле, в Windows происходит что-то подобное, только все это скрыто от пользователя.

Фактически смонтированный раздел становится частью корневой файловой системы и система старается сделать работу со всеми разделами, независимо от их файловых систем, прозрачной. Это значит, что если вы примонтируете участок оперативной памяти или удаленную сетевую папку, то сможете работать с ней в файловом менеджере точно так же, как и с локальным диском.

Например, вы хотите примонтировать флешку. Вы даете системе команду подключить ее в папку /run/media/имя\_пользователя/UUID\_флешки/. Система определяет файловую систему устройства, а затем, используя драйвера ядра подключает ее к указанной папке. Дальше вам остается работать с той папкой, как с любой другой. Больше ни о чем думать не нужно. Когда надумаете извлечь флешку, ее нужно отмонтировать.

## МОНТИРОВАНИЕ ДИСКОВ В LINUX

Обычно, монтированием занимаются специальные сервисы оболочки, но не всегда они доступны. А иногда нужно сделать все вручную, чтобы задать дополнительные опции монтирования или другие параметры. Для монтирования в Linux используется команда mount. Рассмотрим ее параметры:

```
$ mount файл_устройства папка_назначения
```

Или расширенный вариант:

```
$ mount опции -t файловая_система -o опции_монтирования файл_устройства  
папка_назначения
```

Опции задают различные дополнительные особенности работы утилиты. Опция -t необязательна, но она позволяет задать файловую систему, которая будет использована и иногда это очень полезно. С помощью опции -o вы можете задать различные параметры монтирования, например, монтировать только для чтения и т.д. Последних два параметра - это файл устройства, например, /dev/sda1 и папка назначения, например, /mnt.

Перед тем как перейти к рассмотрению примеров работы утилитой, давайте рассмотрим ее основные опции:

- -V - вывести версию утилиты;
- -h - вывести справку;
- -v - подробный режим;
- -a, --all - примонтировать все устройства, описанные в fstab;
- -F, --fork - создавать отдельный экземпляр mount для каждого отдельного раздела;
- -f, --fake - не выполнять никаких действий, а только посмотреть что собирается делать утилита;
- -n, --no-mtab - не записывать данные о монтировании в /etc/mtab;
- -l, --show-labels - добавить метку диска к точке монтирования;
- -c - использовать только абсолютные пути;
- -r, --read-only - монтировать раздел только для чтения;
- -w, --rw - монтировать для чтения и записи;
- -L, --label - монтировать раздел по метке;
- -U, --uuid - монтировать раздел по UUID;
- -T, --fstab - использовать альтернативный fstab;
- -B, --bind - монтировать локальную папку;
- -R, --rbind - перемонтировать локальную папку.

Это не все, но основные опции, которые вам понадобятся во время работы с утилитой. Также, возможно, вы захотите знать список опций монтирования, которые могут быть полезными. Они все перечислены в статье автоматическое монтирование в fstab и писать их еще и здесь нет смысла. А теперь перейдем к примерам и рассмотрим как монтировать диск в linux.

## МОНТИРОВАНИЕ РАЗДЕЛОВ С ПОМОЩЬЮ MOUNT

Монтирование разделов с помощью mount выполняется очень просто. Фактически в большинстве случаев будет достаточно упрощенной версии команды. Например, смонтируем раздел /dev/sdb6 в папку /mnt:

```
sudo mount /dev/sdb6 /mnt/
```

В большинстве случаев вы будете вынуждены выполнять команду mount с правами суперпользователя, если обратное не указано в fstab (опция монтирования users). Вы можете посмотреть информацию о процессе монтирования добавив опцию -v:

```
sudo mount -v /dev/sdb6 /mnt/
```

Если нужно, вы можете указать файловую систему с помощью опции -t:

```
sudo mount -v -t ext4 /dev/sdb6 /mnt
```

Если необходимо примонтировать файловую систему только для чтения, то вы можете использовать опцию -r или опцию монтирования -o ro, результат будет одинаковым:

```
sudo mount -t ext4 -r /dev/sdb6 /mnt  
sudo mount -t ext4 -o ro /dev/sdb6 /mnt
```

Вы можете использовать и другие опции чтобы выполнить монтирование разделов linux, например, указать, что на этом разделе нельзя выполнять программы:

```
sudo mount -t ext4 -o noexec /dev/sdb6 /mnt
```

Обратите внимание, что вы не можете использовать опции uid, gid, fmask для файловых систем ext. Они поддерживаются только в FAT, vFAT, exFAT.

Вы можете использовать не только имена устройств чтобы выполнить монтирование диска в linux. Для этого можно применять UUID или метки, например, монтирование с помощью UUID:

```
sudo mount --uuid="b386d309-05c1-42c8-8364-8d37270b69e0" /mnt
```

Посмотреть uuid для ваших разделов можно с помощью команды:

```
sudo blkid
```

Точно так же вы можете использовать метки. Команда монтирования диска linux будет выглядеть так:

```
sudo mount --label="home" /mnt/
```

Вы можете примонтировать одну папку в другую, для этого используйте опцию --bind

```
sudo mount --bind /mnt/ /media/
```

Возможно, не только монтирование разделов linux, но и монтирование файлов, если они содержат файловую систему, например, образов дисков. Монтирование образа диска linux работает точно так же:

```
sudo mount ~/file.iso /mnt
```

Посмотреть список всех примонтированных устройств можно просто выполнив mount без параметров:

```
mount
```

## РАЗМОНТИРОВАНИЕ УСТРОЙСТВ В LINUX

Когда вы хотите завершить работу с устройством, особенно с флешкой, его нужно размонтировать. Для этого существует утилита `umount`. В качестве параметров она принимает точку монтирования или устройство. Например:

```
sudo umount /mnt
```

Теперь ваше устройство не смонтировано. Но иногда может возникнуть ошибка размонтирования. Система сообщит, что устройство занято: `umount: /mnt: target is busy.`

Проблему можно решить закрыв все программы, которые могут использовать любой файл в этой папке. Какие именно это программы вы можете узнать с помощью команды `lsof`:

```
lsof -w /mnt  
$ lsof -w /dev/sdb6
```

Здесь вы видите всю необходимую информацию, чтобы понять что происходит и что с этим делать. Утилита вывела название программы, ее PID, и даже файл, с которым она работает. Вы можете завершить все программы, а потом снова повторить попытку или используйте опцию `-l`, файловая система будет отключена немедленно, несмотря на то, что она занята:

```
sudo umount -l /mnt
```

## ТИПЫ ФАЙЛОВЫХ СИСТЕМ ДЛЯ LINUX

Операционная система Windows может быть установлена только на файловую систему NTFS, поэтому обычно у пользователей не возникает вопросов какую ФС лучше использовать. Но Linux очень сильно отличается, здесь в ядро системы встроены и могут использоваться несколько файловых систем, каждая из которых оптимизирована для решения определенных задач и лучше подходит именно для них.

## ЖЕСТКИЙ ДИСК И РАЗДЕЛЫ

Обычно в компьютере используется один жесткий диск, но для удобства все доступное пространство разделяется на разделы, в Windows они известны как диски, в Linux же их принято называть разделами. Чтобы операционная система знала сколько разделов есть на диске и их физические границы используется таблица разделов. Она может быть двух типов - GPT или MBR. В этой статье мы не будем рассматривать ее подробно. Скажу только, что там находится метка раздела, его порядковый номер и адрес начала и конца на жестком диске.

## ЧТО ТАКОЕ ФАЙЛОВАЯ СИСТЕМА?

Дальше больше. Чтобы на каждом разделе можно было работать с файлами и каталогами, необходима файловая система. Мы могли бы писать просто содержимое файлов на диск, но нужно еще где-то хранить данные о папках, имена файлов, их размер, адрес на жестком диске, атрибуты доступа. Всем этим занимается файловая система.

От файловой системы зависит очень многое, скорость работы с файлами, скорость записи и даже размер файлов. Также от стабильности файловой системы будет зависеть сохранность ваших файлов.

Файловые системы в Linux используются не только для работы с файлами на диске, но и для хранения данных в оперативной памяти или доступа к конфигурации ядра во время работы системы. Дальше мы рассмотрим типы файловых систем Linux, включая специальные файловые системы.

## ОСНОВНЫЕ ФАЙЛОВЫЕ СИСТЕМЫ

Каждый дистрибутив Linux позволяет использовать одну из этих файловых систем, каждая из них имеет свои преимущества и недостатки. Все они включены в ядро и могут использоваться в качестве корневой файловой системы. Давайте рассмотрим каждую из них более подробно.

- Ext2, Ext3, Ext4 или Extended Filesystem - это стандартная файловая система для Linux. Она была разработана еще для Minix. Она самая стабильная из всех существующих, кодовая база изменяется очень редко и эта файловая система содержит больше всего функций. Версия ext2 была разработана уже именно для Linux и получила много улучшений. В 2001 году вышла ext3, которая добавила еще больше стабильности благодаря использованию журналирования. В 2006 была выпущена версия ext4, которая используется во всех дистрибутивах Linux до сегодняшнего дня. В ней было внесено много улучшений, в том числе увеличен максимальный размер раздела до одного экзабайта.
- JFS или Journaled File System была разработана в IBM для AIX UNIX и использовалась в качестве альтернативы для файловых систем ext. Сейчас она используется там, где необходима высокая стабильность и минимальное потребление ресурсов. При разработке файловой системы ставилась цель создать максимально эффективную файловую систему для многопроцессорных компьютеров. Так же как и ext, это журналируемая файловая система, но в журнале хранятся только метаданные, что может привести к использованию старых версий файлов после сбоев.
- ReiserFS - была разработана намного позже, в качестве альтернативы ext3 с улучшенной производительностью и расширенными возможностями. Она была разработана под руководством Ганса Райзера и поддерживает только Linux. Из особенностей можно отметить динамический размер блока, что позволяет упаковывать несколько небольших файлов в один блок, что предотвращает фрагментацию и улучшает работу с небольшими файлами. Еще одно преимущество - в возможности изменять размеры разделов на лету. Но минус в некоторой нестабильности и риске потери данных при отключении энергии. Раньше ReiserFS применялась по умолчанию в SUSE Linux, но сейчас разработчики перешли на Btrfs.
- XFS - это высокопроизводительная файловая система, разработанная в Silicon Graphics для собственной операционной системы еще в 2001 году. Она изначально была рассчитана на файлы большого размера, и поддерживала диски до 2 Терабайт. Из преимуществ файловой системы можно отметить высокую скорость работы с большими файлами, отложенное выделение места, увеличение разделов на лету и незначительный размер служебной информации.
- XFS - журналируемая файловая система, однако в отличие от ext, в журнал записываются только изменения метаданных. Она используется по умолчанию в дистрибутивах на основе Red Hat. Из недостатков - это невозможность уменьшения размера, сложность восстановления данных и риск потери файлов при записи, если будет неожиданное отключение питания, поскольку большинство данных находится в памяти.

- Btrfs или B-Tree File System - это совершенно новая файловая система, которая сосредоточена на отказоустойчивости, легкости администрирования и восстановления данных. Файловая система объединяет в себе очень много новых интересных возможностей, таких как размещение на нескольких разделах, поддержка подтомов, изменение размера не лету, создание мгновенных снимков, а также высокая производительность. Но многими пользователями файловая система Btrfs считается нестабильной. Тем не менее, она уже используется как файловая система по умолчанию в OpenSUSE и SUSE Linux.

Другие файловые системы, такие как NTFS, FAT, HFS могут использоваться в Linux, но корневая файловая система linux на них не устанавливается, поскольку они для этого не предназначены.

## СПЕЦИАЛЬНЫЕ ФАЙЛОВЫЕ СИСТЕМЫ

Ядро Linux использует специальные файловые системы, чтобы предоставить доступ пользователю и программам к своим настройкам и информации. Наиболее часто вы будете сталкиваться с такими вариантами:

- tmpfs позволяет размещать любые пользовательские файлы в оперативной памяти компьютера. Достаточно создать блочное устройство нужного размера, затем подключить его к папке, и вы можете писать файлы в оперативную память.
- procfs - по умолчанию смонтирована в папку proc и содержит всю информацию о запущенных в системе процессах, а также самом ядре.
- sysfs - с помощью этой файловой системы вы можете задавать различные настройки ядра во время выполнения.

## ВИРТУАЛЬНЫЕ ФАЙЛОВЫЕ СИСТЕМЫ

Не все файловые системы нужны в ядре. Существуют некоторые решения, которые можно реализовать и в пространстве пользователя. Разработчики ядра создали модуль FUSE ( filesystem in userspace), который позволяет создавать файловые системы в пространстве пользователя. К виртуальным файловым системам можно отнести ФС для шифрования и сетевые файловые системы.

- EncFS - файловая система, которая шифрует все файлы и сохраняет их в зашифрованном виде в нужную директорию. Получить доступ к расшифрованным данным можно только примонтировав файловую систему.
- Aufs (AnotherUnionFS) - позволяет объединять несколько файловых систем (папок) в одну общую.
- NFS (Network Filesystem) - позволяет примонтировать файловую систему удаленного компьютера по сети.

Таких файловых систем очень много, и мы не будем перечислять все их в данной статье. Есть даже очень экзотические варианты, обратите внимание на проект Plfs.