

ФАЙЛОВАЯ СИСТЕМА LINUX

Любая операционная система поддерживает несколько файловых систем, например, Windows поддерживает файловые системы FAT , FAT32 , NTFS , ISO9660 , UDFи др. Linux — тоже не исключение. Однако нужно различать файловые системы, которые могут использоваться в качестве корневой файловой системы, и файловые системы, которые просто поддерживает ОС, но которые не используются для установки Linux.

В качестве корневой файловой системы применяются следующие файловые системы:

ФАЙЛОВАЯ СИСТЕМА LINUX

- **ext** — первая файловая система Linux, использовалась в ранних версиях Linux;
- **ext2** — стандартная, но уже устаревшая файловая система Linux. Долгое время использовалась практически во всех дистрибутивах по умолчанию, но была заменена файловой системой ext3;

ФАЙЛОВАЯ СИСТЕМА LINUX

- **ext3** — модифицированная версия файловой системы ext2, но с поддержкой журнала, существенно повышающего надежность файловой системы.

Максимальный размер раздела с файловой системой ext3 — 4 Тбайт, хотя ядро Linux версии 2.6 поддерживает максимальный размер раздела до 16 Тбайт. Максимальный размер файла — 1 Тбайт.

ФАЙЛОВАЯ СИСТЕМА LINUX

- **ext4** — новейшая файловая система Linux. Поддержка ext4 как стабильной файловой системы появилась в ядре Linux версии 2.6.28 — это самая новая версия ядра на момент написания этих строк. Если сравнивать эту файловую систему с ext3, то производительность и надежность новой файловой системы существенно увеличена, а максимальный размер раздела теперь равен 1024 Пбайт (1 Эбайт) . Максимальный размер файла больше 2 Тбайт.

Петабайт (Пбайт) — единица количества информации, равная 1024 Тбайт.

Эксабайт (Эбайт) — единица количества информации, равная 1024 Пбайт.

ФАЙЛОВАЯ СИСТЕМА LINUX

- **ReiserFS** — основная особенность этой файловой системы заключается в хранении в одном блоке нескольких маленьких файлов. Например, если у вас размер блока 4 Кбайт, то в него поместится до четырех файлов по одному килобайту каждый. Если у вас много маленьких файлов, то такая файловая система — просто находка, ведь она позволяет экономить дисковое пространство. Однако с большими файлами эта файловая система работает медленно, потом она чувствительна к сбоям и ее нужно регулярно дефрагментировать. И еще: если вы решили использовать эту файловую систему, обзаведитесь источником бесперебойного питания, который предотвратит потерю данных в случае отключения электричества

ФАЙЛОВАЯ СИСТЕМА LINUX

- **JFS** — разработка IBM, обладает высокой производительностью, но оптимизирована под сервер баз данных, поскольку размер блока небольшой — от 512 байт до 4 Кбайт. Если вам приходится работать с большими файлами, например с видео, то файловая система — это не очень удачный выбор.
- **XFS** — обладает относительно высокой производительностью — она быстрее, чем ext3, ReiserFS и JFS, но медленнее, чем ext4. Устанавливает большой размер блока — до 64 Кбайт, что позволяет ее использовать на графических станциях для обработки видео.

ФАЙЛОВАЯ СИСТЕМА LINUX

Ресурс Phoronix (www.phoronix.com) провел тестирование новой файловой системы на SSD-накопителе (такие накопители устанавливаются в современные нетбуки). Результат, как говорится, на лицо (рис. 1): ext4 почти в два раза превзошла файловые системы ext3, XFS, JFS и ReiserFS.

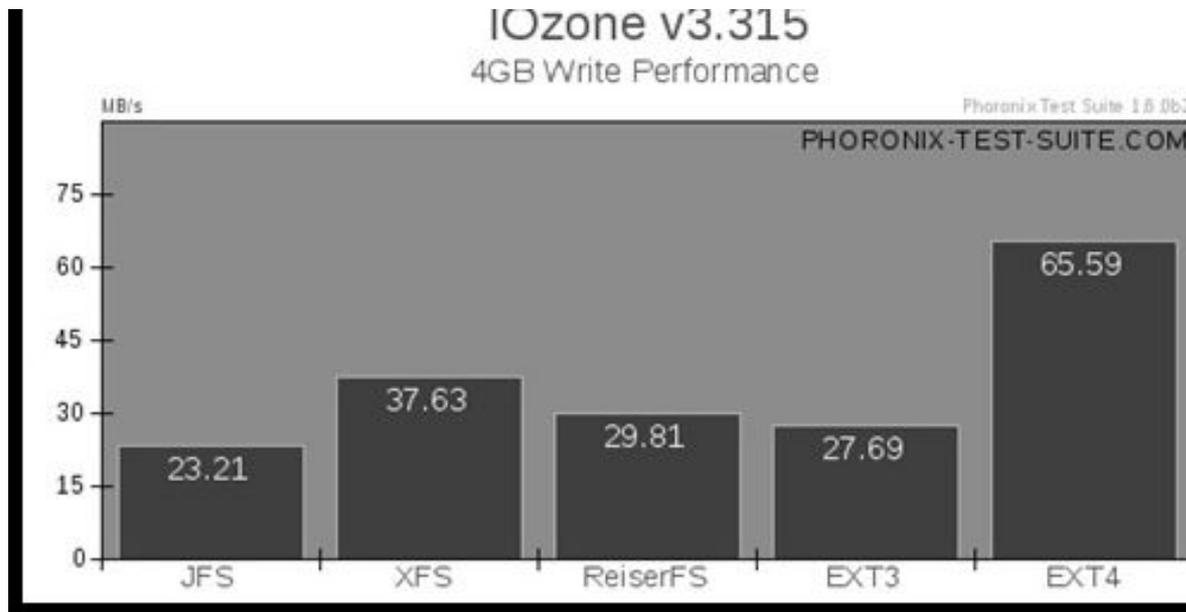


Рис. 1. Тест производительности файловых систем, проведенный ресурсом Phoronix

ФАЙЛОВАЯ СИСТЕМА LINUX

Какую файловую систему выбрать сегодня?

Если любите экспериментировать и не планируете хранить на своем компьютере ничего важного, можно выбрать ext4. Ведь она обеспечивает высокую производительность. Но совсем недавно она была нестабильной, поэтому если у вас хранится что-то важное, тогда лучше выбрать ext3.

КОРНЕВАЯ ФАЙЛОВАЯ СИСТЕМА LINUX, СТАНДАРТНЫЕ КАТАЛОГИ

При установке Linux на выбранном вами разделе жесткого диска создается корневая файловая система Linux. Корневая файловая система содержит набор стандартных каталогов и утилит, без которых невозможна работа Linux.

Корневая файловая система обозначается как **/**.

Обратите внимание: в Linux используется прямой слеш, а не обратный (\), как в Windows. Полный путь к файлу обязательно начинается с корневой файловой системы. Полный путь к файлу report.doc, который находится в домашнем каталоге пользователя den:

`/home/den/report.doc`.

КОРНЕВАЯ ФАЙЛОВАЯ СИСТЕМА LINUX, СТАНДАРТНЫЕ КАТАЛОГИ

Основные каталоги корневой файловой системы

/ - Корневой каталог;

/bin - Содержит стандартные утилиты Linux;

/boot - Содержит конфигурационные файлы загрузчика GRUB , образы ядра, файлы Initrd;

(Initrd — *диск в оперативной памяти для начальной инициализации* – временная файловая система, используемая ядром Linux при начальной загрузке. Initrd обычно используется для начальной инициализации перед монтированием «настоящих» файловых систем.)

КОРНЕВАЯ ФАЙЛОВАЯ СИСТЕМА LINUX, СТАНДАРТНЫЕ КАТАЛОГИ

/dev - Содержит файлы устройств;

/etc - Содержит конфигурационные файлы операционной системы и всех сетевых служб. Данный каталог подобен реестру Windows, но в Windows общесистемные настройки хранятся в одном большом бинарном файле, а в Linux настройки хранятся в разных конфигурационных файлах, которые можно редактировать обычным текстовым редактором;

КОРНЕВАЯ ФАЙЛОВАЯ СИСТЕМА LINUX, СТАНДАРТНЫЕ КАТАЛОГИ

/home - Содержит домашние каталоги пользователей. В домашних каталогах пользователей хранятся пользовательские файлы, а также пользовательские настройки различных программ;

/lib - Здесь находятся различные библиотеки и модули ядра;

/misc - В данном каталоге может быть все, что угодно;

КОРНЕВАЯ ФАЙЛОВАЯ СИСТЕМА LINUX, СТАНДАРТНЫЕ КАТАЛОГИ

/mnt - Обычно в этом каталоге содержатся точки монтирования. О монтировании файловых систем мы тоже поговорим отдельно;

/proc - Это не совсем обычный каталог, это каталог псевдофайловой системы `procfs`, которая используется для предоставления информации о процессах;

/root - Каталог пользователя `root` (пользователь с максимальными полномочиями);

КОРНЕВАЯ ФАЙЛОВАЯ СИСТЕМА LINUX, СТАНДАРТНЫЕ КАТАЛОГИ

/sbin - Набор утилит для системного администрирования, запускать эти утилиты имеет право только root;

/tmp - “Мусорка”, т.е. каталог, в котором хранятся временные файлы. Linux, в отличие от Windows, следит за чистотой и регулярно очищает этот каталог;

КОРНЕВАЯ ФАЙЛОВАЯ СИСТЕМА LINUX, СТАНДАРТНЫЕ КАТАЛОГИ

/usr - Содержит пользовательские программы. По размеру это один из самых больших каталогов файловой системы. В этот каталог устанавливаются практически все программы. Также в этом каталоге находятся вспомогательные файлы, необходимые для работы установленных программ. Грубо, конечно, но данный каталог можно сравнить с каталогом Program Files в Windows;

/var - Данные системы, которые постоянно изменяются, например, очередь печати, почтовые ящики и т.д.

ИМЕНА ФАЙЛОВ

В **Windows** имя файла состоит из двух частей — собственно из имени файла и расширения, позволяющего установить тип файла.

В **Linux** вообще нет такого понятия, как расширение файла.

Если в **Windows** в имени `index.html` последние четыре символа — это расширение файла, то в **Linux** — это просто четыре символа после последней точки в имени файла.

ИМЕНА ФАЙЛОВ

Относительно имен файлов вам нужно знать еще следующие факты:

- Имя файла не может содержать символы / \ ? < > * " |
- Максимальная длина имени файла — 254 символа
- Linux чувствительна к регистру букв в имени файла, например, INDEX.html и Index.html — это два разных файла
- Желательно не использовать в именах файлов русские буквы. Linux нормально работает с такими файлами, но могут возникнуть проблемы при передаче таких файлов в другую операционную систему, например в Windows
- Разделение элементов пути к файлу осуществляется с помощью прямого слеша (/), а не обратного (\), как в Windows

ФАЙЛЫ УСТРОЙСТВ

Абстракция — это великая сила. Возьмем, например, DOS. Разные разделы жесткого диска в DOS представляются как отдельные диски (в DOS они называются логическими дисками). В Windows (NT) то же самое — открываем окно Мой компьютер и видим несколько пиктограмм жестких дисков. Каждый логический диск представляется как отдельный дисковый накопитель. Для пользователя нет разницы, куда он записывает данные — на отдельный физический диск или на раздел физического диска.

Linux по уровню абстракции опередила Windows. В Linux каждое отдельное устройство представляется в виде файла файловой системы, и к устройству вы можете обращаться как к обычному файлу! Файлы устройств хранятся в каталоге /dev.

ФАЙЛЫ УСТРОЙСТВ

/dev/modem - Файл модема, но обычно это ссылка или на устройство `ttySn`, или на `ttyUSBn`;

/dev/ttySn - где `n` — номер файла последовательного порта, `ttyS0=COM1`, `ttyS1=COM2`;

/dev/ttyUSBn - где `n` — номер, обычно используется в случае подключения USB-модема

/dev/scdN - где `N` — номер, имя файла привода CD/DVD, `N` — номер привода

/dev/srN, где `N` — номер Ссылка на устройство `/dev/scdN`

/dev/fdN, где `N` — номер Файл дисководов гибких дисков:

/dev/fd0 — A:, **/dev/fd1** — B:

КОМАНДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ

cat имя _ файла - Просмотр текстового файла. Обычно используется в паре с командой `more` для организации постраничного вывода: `cat <имя файла> | more`

tac имя _ файла Вывод файла в обратном порядке. Полезна для просмотра файлов журналов, в которых самая свежая информация находится в конце файла. Для организации постраничного просмотра нужно использовать программу `more` или `less`: `tac <имя файла> | less`

touch имя _ файла Создает пустой файл

КОМАНДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ

echo строка Выводит указанную строку. С помощью этой команды можно добавить строку в имеющийся файл или же создать новый файл с заданной строкой:

echo строка > новый _ файл

echo строка >> существующий _ файл

В первом случае, если файл существует, то он будет перезаписан

cp источник назначение Копирует файл источник в файл назначение. Если файл назначение существует, то программа спросит вас, нужно ли его перезаписать или нет

КОМАНДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ

mv источник назначение Перемещает файл источник в файл назначение. Если файл назначение существует, то программа спросит вас, нужно ли его перезаписать или нет. Данную команду можно использовать также для переименования файла

less файл Постраничный вывод файла

locate файл Выполняет поиск файла

In файл ссылка Используется для создания ссылки на указанный файл. Ссылка — это еще одно имя файла. (Файл нельзя удалить, если на него указывает хотя бы одна ссылка)

КОМАНДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ

rm *файл* Удаляет файл

which *исполнимый _ файл* Используется для поиска программы в каталогах, указанных в переменной окружения PATH

chmod *режим файла* Изменение прав доступа к файлу или каталогу. Подробнее см. `man chmod`

chown *пользователь файла* Изменение владельца файла. Подробнее см. `man chown`

КОМАНДЫ ДЛЯ РАБОТЫ С КАТАЛОГАМИ

cd *каталог* Изменение каталога

mkdir *каталог* Создание каталога

rmdir *каталог* Удаление пустого каталога

rm -r *каталог* Рекурсивное удаление непустого каталога (будут удалены все подкаталоги заданного каталога и сам каталог)

ls *каталог* Вывод содержимого каталога

КОМАНДЫ ДЛЯ РАБОТЫ С КАТАЛОГАМИ

три специальных имени каталогов:

- ~ — домашний каталог пользователя;
- — текущий каталог;
- .. — родительский каталог.

ПРАВА ДОСТУПА К ФАЙЛАМ

В Linux всех пользователей можно разделить на три большие группы:

- **Владелец.** К этой группе принадлежит всего один пользователь, создавший файл или каталог.
- **Группа.** Члены группы, к которой принадлежит владелец. Например, может быть создана группа `users`, к которой будут принадлежать пользователи `ivanov`, `petrov` и `sidorov`.
- **Прочие.** Все остальные пользователи, которые не являются владельцем файла и которые не входят в группу, к которой принадлежит владелец файла.

ПРАВА ДОСТУПА К ФАЙЛАМ

Вот что нужно помнить о правах доступа в Linux.

- **Вы, владелец файла, можете назначить права доступа к собственным файлам и каталогам для каждой группы пользователей.**
- **Пользователь может создавать файлы только в пределах домашнего каталога. Доступ к остальным каталогам файловой системы ограничен (как правило, только чтением, без возможности создания новых файлов и изменения уже имеющихся).**
- **Изменить права доступа к файлу может владелец или пользователь root.**
- **Пользователь root обладает максимальными привилегиями, поэтому он может получить полный доступ к любому файлу и каталогу.**

ПРАВА ДОСТУПА К ФАЙЛАМ

как установить права с помощью
командной строки Linux?

в Linux существует три права доступа:

чтение (r — от *read*),

запись (w — от *write*),

выполнение (x от *eXecute*).

ПРАВА ДОСТУПА К ФАЙЛАМ

`ls -l <имя _ файла>`

`-rw-r--r-- 1 den den-group 6051 Ноя 28 14:44 <имя _ файла>`



Права доступа к
файлу

команды установки прав доступа:

`chmod 666 <файл>`

`chmod 777 <каталог>`

ПРАВА ДОСТУПА К ФАЙЛАМ

Посмотрим на набор прав:

rw- r -- r--

Теперь заменим символы на 1, а тире — на 0. Получим:

110 100 100

Табличку восьмеричных значений.

Восьмеричная система счисления

Двоичная система

Восьмеричная

000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Подставим полученные двоичные значения в таблицу и в итоге получим набор прав доступа 644 (владельцу разрешено чтение и запись, всем остальным — только чтение).

ПРАВА ДОСТУПА К ФАЙЛАМ

Набор 666 соответствует правам доступа:

rw- rw- rw-

Это самые небезопасные права доступа, позволяющие всем желающим делать с файлом все, что они хотят (вплоть до удаления). Поэтому права 666 лучше не устанавливать для файлов, содержащих что-то важное. Для таких файлов больше подойдут права 600, разрешающие операции с файлом только владельцу.

Права 777 обычно устанавливаются для каталогов. Это то же, что и 666, но для каталога.

Команда `chown` используется для изменения владельца файла. Например:

`chown user report.txt`

После выполнения этой команды пользователь **user** станет владельцем файла `report.txt`. Вы можете изменить владельца файла только для файлов, владельцем которых вы являетесь. Пользователь `root` может изменить владельца для любого файла.

ЖЕСТКИЕ И СИМВОЛИЧЕСКИЕ ССЫЛКИ

В Linux вы можете создавать ссылки на файл (или каталог). По сути, один и тот же файл может храниться в файловой системе под разными именами — **допускается неограниченное количество ссылок** на файл.

Ссылки бывают жесткими и символическими. Жесткие ссылки “жестко” привязаны к файлу: **нельзя удалить файл**, пока на него указывает хотя бы одна ссылка. Если на файл указывают только **символьные ссылки, вы можете его удалить**, потом ссылки окажутся битыми — они будут ссылаться “в никуда”.

Жесткие ссылки не могут указывать на файл, который физически находится за пределами файловой системы, например, находится на другом разделе.

Для создания ссылок используется команда **ln**, ее синтаксис следующий:

ln [-s] <файл ссылка>

Если указан параметр **-s**, то будет создана символическая ссылка:

ln report.doc hard _ link.doc

- жесткая ссылка

ln -s report.doc sym _ link

- символическая ссылка

МОНТИРОВАНИЕ

Команда **mount**

Операция монтирования: на вашем жестком диске, кроме Linux-разделов, есть разделы других операционных систем (например, Windows).

Для того чтобы получить доступ к файлам, находящимся на этих разделах, нужно примонтировать данные разделы к корневой файловой системе.

Точкой монтирования называется каталог, к которому подмонтирована другая файловая система.

Через точку монтирования осуществляется доступ к файлам и каталогам подмонтированного раздела.

МОНТИРОВАНИЕ

Монтирование выполняется с помощью команды `mount` (ее имеет право вводить только пользователь `root`, поэтому запуск команды `mount` должен осуществляться через команду `sudo`):

```
sudo mount <раздел> <точка_монтирования>
```

Например:

```
sudo mount /dev/sda1 /mnt/disk_с
```

Точка монтирования (каталог `/mnt/disk_с` в нашем случае) должна существовать на момент монтирования.

МОНТИРОВАНИЕ

Приведенная выше команда монтирует первый раздел на первом IDE-диске (обычно это диск C:) к каталогу /mnt/disk_c. Далее вы можете обращаться к файлам и каталогам диска C:

через каталог /mnt/disk_c:

ls /mnt/disk_c

Обычно точки монтирования принято создавать в каталоге /mnt, но это не принципиально: при желании вы можете подмонтировать раздел к любому другому каталогу.

МОНТИРОВАНИЕ

Приведенная выше команда монтирует первый раздел на первом IDE-диске (обычно это диск C:) к каталогу `/mnt/disk_c`. Далее вы можете обращаться к файлам и каталогам диска C:

через каталог `/mnt/disk_c`:

`ls /mnt/disk_c`

Обычно точки монтирования принято создавать в каталоге `/mnt`, но это не принципиально: при желании вы можете подмонтировать раздел к любому другому каталогу.

ЖУРНАЛИРУЕМАЯ ФАЙЛОВАЯ СИСТЕМА

Файловая система ext3 , которая используется в любом современном дистрибутиве по умолчанию, имеет три режима работы журнала:

- **ordered** — в данном режиме в журнал заносится информация только об изменении служебных данных файловой системы. Режим используется по умолчанию, это компромисс между отказоустойчивостью и скоростью;
- **journal** — используйте, если хотите минимизировать потери данных в случае сбоя. В этом режиме в журнал записываются и пользовательские данные, и служебные данные файловой системы. Самый медленный режим;
- **writeback** — самый быстрый, но самый бесполезный, толку от него мало.

ЖУРНАЛИРУЕМАЯ ФАЙЛОВАЯ СИСТЕМА

Установить режим работы журнала можно с помощью опции файловой системы **data**

(в файле `/etc/fstab`):

раздел точка _ монтирования ext 3 data=режим 1 1

Режим работы журнала можно задавать только для файловой системы ext3!

Пример:

`/dev/sda5 / ext3 defaults,data=journal 1 1`

ПРОВЕРКА ФАЙЛОВОЙ СИСТЕМЫ

Для проверки файловой системы используется программа **fsck** (сокращение от *filesystem check*). *С помощью данной программы вы можете проверять не только файловые системы Linux, но и Windows-разделы.*

Единственное требование — перед началом проверки проверяемая файловая система должна быть размонтирована.

Поэтому последовательность действий для проверки файловой системы будет следующей:

```
sudo umount <имя _ раздела>
```

```
sudo <имя_раздела>
```

Например, для проверки /dev/sda1 можно использовать команды:

```
sudo umount /dev/sda1
```

```
sudo fsck /dev/sda1
```