

Концепции файловой системы Linux

Стелина Петрити ; НПИбд-02-21 ; 1032205421

1. Основные функции файловой системы
2. Структура каталогов
3. Верхний уровень иерархии файловой системы Linux.
4. Типы файловых систем

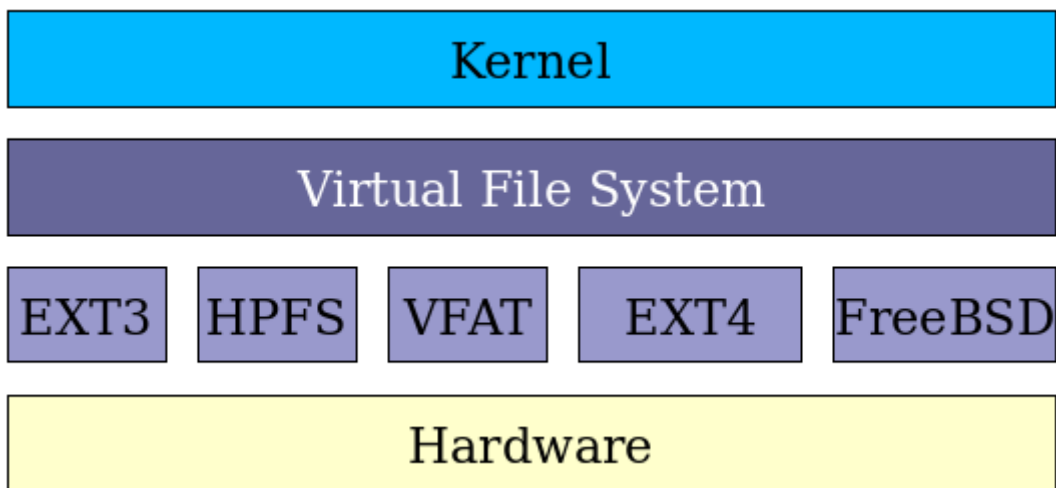
1. Основные функции файловой системы

Дисковое хранилище - это необходимость, которая влечет за собой некоторые интересные и неизбежные детали. Очевидно, что файловая система предназначена для обеспечения пространства для энергонезависимого хранения данных; это ее конечная функция. Однако есть много других важных функций, которые вытекают из этого требования.

Все файловые системы должны предоставлять пространство имен, то есть методологию именования и организации. Это определяет, как файл может быть назван, в частности, длину имени файла и подмножество символов, которые могут быть использованы для имен файлов из общего набора доступных символов. Он также определяет логическую структуру данных на диске, например, использование каталогов для организации файлов вместо того, чтобы просто объединять их все в один огромный конгломерат файлов. Файловые системы также требуют Интерфейса прикладного программирования (API), который обеспечивает доступ к вызовам системных функций, которые управляют объектами файловой системы, такими как файлы и каталоги. API-интерфейсы обеспечивают выполнение таких задач, как создание, перемещение и удаление файлов. Он также предоставляет алгоритмы, которые определяют такие вещи, как место размещения файла в файловой системе. Такие алгоритмы могут учитывать такие цели, как скорость или минимизация фрагментации диска.

Современные файловые системы также предоставляют модель безопасности, которая представляет собой схему определения прав доступа к файлам и каталогам. Модель безопасности файловой системы Linux помогает гарантировать, что пользователи имеют доступ только к своим собственным файлам, а не к файлам других пользователей или самой операционной системы.

Последним строительным блоком является программное обеспечение, необходимое для реализации всех этих функций. Linux использует двухкомпонентную программную реализацию как способ повышения эффективности как системы, так и программиста.



Первой частью этой реализации, состоящей из двух частей, является виртуальная файловая система Linux. Эта виртуальная файловая система предоставляет единый набор команд для ядра и разработчиков для доступа ко всем типам файловых систем. Программное обеспечение виртуальной файловой системы вызывает определенный драйвер устройства, необходимый для взаимодействия с различными типами файловых систем. Драйверы устройств, зависящие от файловой системы, являются второй частью реализации. Драйвер устройства интерпретирует стандартный набор команд файловой системы в команды, специфичные для типа файловой системы на разделе или логическом томе.

2. Структура каталогов

В Linux и многих других операционных системах каталоги могут быть структурированы в виде древовидной иерархии. Структура каталогов Linux четко определена и задокументирована в Стандарте иерархии файловой системы Linux (FHS).

Ссылки на эти каталоги при доступе к ним выполняются с помощью последовательно более глубоких имен каталогов, соединенных косой чертой (/), таких как `/var/log` и `/var/spool/mail`. Они называются путями.

Корневая файловая система - это каталог верхнего уровня файловой системы. Он должен содержать все файлы, необходимые для загрузки системы Linux, прежде чем будут смонтированы другие файловые системы. Он должен включать все необходимые исполняемые файлы и библиотеки, необходимые для загрузки остальных файловых систем. После загрузки системы все остальные файловые системы монтируются в стандартные, четко определенные точки монтирования в качестве подкаталогов корневой файловой системы.

3. Верхний уровень иерархии файловой системы Linux.

/bin Каталог `/bin` содержит пользовательские исполняемые файлы.

/boot- Содержит статический загрузчик, исполняемый файл ядра и конфигурационные файлы, необходимые для загрузки компьютера с Linux.

/dev- Этот каталог содержит файлы устройств для каждого аппаратного устройства, подключенного к системе. Это не драйверы устройств, скорее это файлы, которые представляют каждое устройство на компьютере и облегчают доступ к этим устройствам.

/etc- Содержит файлы конфигурации локальной системы для главного компьютера.

/home- Хранилище домашнего каталога для пользовательских файлов. У каждого пользователя есть подкаталог в /home.

/lib- Содержит файлы общей библиотеки, необходимые для загрузки системы.

/media- Место для установки внешних съемных носителей, таких как флэш-накопители USB, которые могут быть подключены к хосту.

/mnt- Временная точка монтирования для обычных файловых систем (например, для несъемных носителей), которую можно использовать, пока администратор восстанавливает файловую систему или работает с ней.

/opt- Необязательные файлы, такие как поставляемые поставщиком прикладные программы, должны быть расположены здесь.

/root- Это не корневая файловая система (/). Это домашний каталог для пользователя root.

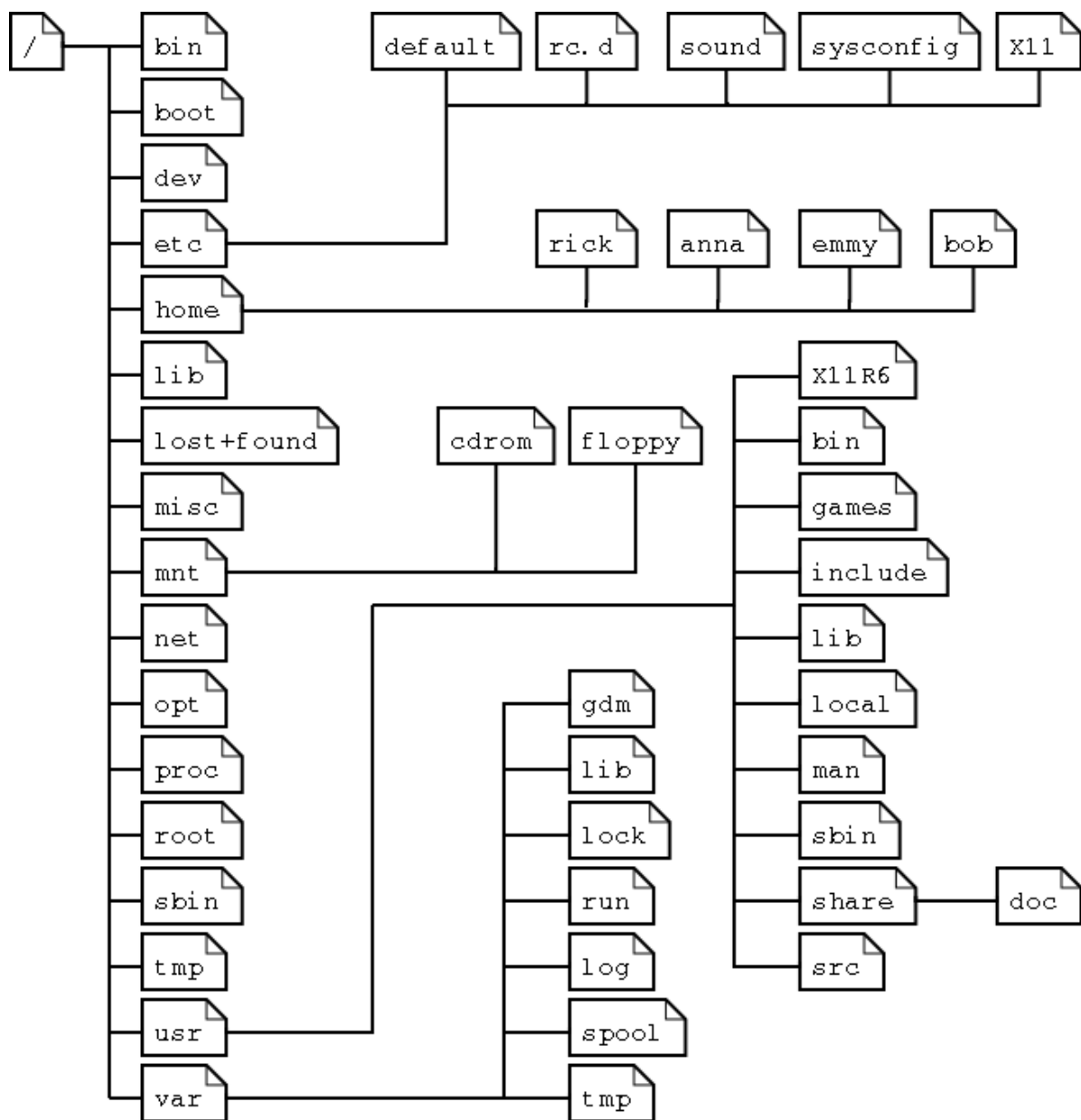
Системные двоичные файлы /sbin- Это исполняемые файлы, используемые для системного администрирования.

/Временный каталог /temp- Используется операционной системой и многими программами для хранения временных файлов. Пользователи также могут временно хранить здесь файлы. Обратите внимание, что файлы, хранящиеся здесь, могут быть удалены в любое время без предварительного уведомления.

/user- Это общедоступные файлы, доступные только для чтения, включая исполняемые двоичные файлы и библиотеки, множество файлов и другие типы документации.

/var- Здесь хранятся файлы переменных данных. Это может включать в себя такие вещи, как файлы журналов, MySQL и другие файлы баз данных, файлы данных веб-сервера, почтовые ящики электронной почты и многое другое.

Каталоги /media и /mnt- являются частью корневой файловой системы, но они никогда не должны содержать никаких данных. Скорее всего, это просто временные точки монтирования.



1. Расположение файловой системы Linux

4. Типы файловых систем

Linux поддерживает чтение около 100 типов разделов; он может создавать и записывать только в некоторые из них. Но возможно — и очень часто — монтировать файловые системы разных типов в одной и той же корневой файловой системе. В этом контексте мы говорим о файловых системах с точки зрения структур и метаданных, необходимых для хранения пользовательских данных и управления ими на разделе жесткого диска или логического тома. Полный список типов разделов файловой системы, распознаваемых командой Linux `fdisk`, приведен здесь, чтобы вы могли почувствовать высокую степень совместимости Linux с очень многими типами систем. Основная цель поддержки возможности чтения такого большого количества типов разделов - обеспечить совместимость и, по крайней мере, некоторую совместимость с файловыми системами других компьютерных систем. Варианты, доступные при создании новой файловой системы с помощью Fedora, показаны в следующем списке.

- btrfs
- **cramfs**
- **ext2**

- **ext3**
- **ext4**
- fat
- gfs2
- hfsplus
- minix
- **msdos**
- ntfs
- reiserfs
- **vfat**
- Xfs

CentOS 6 поддерживает создание только тех файловых систем, которые выделены жирным шрифтом в приведенном выше списке.

Вывод:

В этой презентации я рассказывал о функции основной файловой системы, о структуре каталогов, которая представляла собой древовидную иерархию, а ссылки на каталоги при обращении к ним выполняются с использованием последовательно более глубоких имен каталогов, соединенных косой чертой (/), таких как /var/log и /var/spool/mail, которые называются путями. Мы также объяснили верхний уровень иерархии файловой системы Linux, которым являются /home, /bin, /usr, /var и т.д., А также о типах файловых систем, которые являются ext2, ext3, ext4 и т.д. Итак, чтобы покончить с этим, файловые системы с точки зрения структур и метаданных, необходимых для хранения пользовательских данных и управления ими на разделе жесткого диска или логическом томе.

Источник

<https://opensource.com/life/16/10/introduction-linux-file-systems>

<https://likegeeks.com/linux-file-system/>

https://tldp.org/LDP/intro-linux/html/sect_03_01.html