

Лабораторная работа №3. Организация файловой системы

Файл

Файл -- это понятие, привычное любому пользователю компьютера. Для пользователя каждый файл -- это отдельный предмет, у которого есть начало и конец и который отличается от всех остальных файлов именем и расположением ("как называется" и "где лежит"). Как и любой предмет, файл можно создать, переместить и уничтожить, однако без внешнего вмешательства он будет сохраняться неизменным неопределенно долгое время. Файл предназначен для хранения данных любого типа -- текстовых, графических, звуковых, исполняемых программ и многого другого. Аналогия файла с предметом позволяет пользователю быстро освоиться при работе с данными в операционной системе.

Для операционной системы Linux файл -- не менее важное понятие, чем для её пользователя: все данные, хранящиеся на любых носителях, обязательно находятся внутри какого-нибудь файла, в противном случае они просто недоступны ни для операционной системы, ни для её пользователей. Более того, все устройства, подключённые к компьютеру (начиная клавиатурой и заканчивая любыми внешними устройствами, например, принтерами и сканерами) Linux представляет как файлы (так называемые файлы-дырки). Конечно, файл, содержащий обычные данные, сильно отличается от файла, предназначенного для обращения к устройству, поэтому в Linux определены несколько различных типов файлов. В основном пользователь имеет дело с файлами трёх типов: обычными файлами, предназначенными для хранения данных, каталогами и файлами-ссылками, именно о них и пойдёт речь на данном занятии, о файлах других типов мы познакомимся на следующих занятиях.

файл	Отдельная область данных на одном из носителей информации, у которой есть собственное имя.
------	--

Система файлов: каталоги

Файловая система с точки зрения пользователя -- это "пространство", в котором размещаются файлы, наличие файловой системы позволяет определить не только "как называется файл", но и "где он находится". Различать файлы только по имени было бы слишком неэффективным: про каждый файл приходилось бы помнить, как он называется и при этом заботиться о том, чтобы имена никогда не повторялись. Более того, необходим механизм, позволяющий работать с группами тематически связанных между собой файлов (например, компонентов одной и той же программы или разных главы одной диссертации). Иначе говоря, файлы нужно *систематизировать*.

файловая система	Способ хранения и организации доступа к данным на информационном носителе или его разделе. Классическая файловая система имеет иерархическую структуру, в которой файл однозначно определяется полным путём к нему.
------------------	---

Linux может работать с различными типами файловых систем, которые различаются списком поддерживаемых возможностей, производительностью в разных ситуациях, надёжностью и другими признаками. Подробнее о работе Linux с разными файловыми системами речь пойдёт в лекции [Работа с внешними устройствами](#). В этой лекции будут описаны возможности файловой системы Ext2/Ext3, на сегодня de facto стандартной файловой системы для Linux.

Большинство современных файловых систем (но не все!) используют в качестве основного организационного принципа каталоги. Каталог -- это список ссылок на файлы или другие каталоги. Принято говорить, что каталог *содержит* в себе файлы или другие каталоги, хотя в действительности он только *ссылается* на них, физическое размещение данных на диске обычно никак не связано с размещением каталога. Каталог, на который есть ссылка в данном каталоге, называется подкаталогом или вложенным каталогом. Каталог в файловой системе более всего напоминает библиотечный каталог, содержащий ссылки на объединённые по каким-то признакам книги и другие разделы каталога (файлы и подкаталоги). Ссылка на один и тот же файл может

содержаться в нескольких каталогах одновременно, это может сделать доступ к файлу более удобным. В файловой системе Ext2 каждый каталог -- это отдельный файл особого типа ("d", от англ. "directory"), отличающийся от обычного файла с данными: в нём могут содержаться только ссылки на другие файлы и каталоги.

В файловой системе Linux нет *папок* и *документов*. Есть каталоги и файлы, возможности которых куда шире.

Довольно часто вместо термина каталог можно встретить папка (англ. folder). Этот термин хорошо вписывается в представление о файлах как о предметах, которые можно раскладывать по папкам, однако часть возможностей файловой системы, которая противоречит этому представлению, таким образом затемняется. В частности, с термином "папка" плохо согласуется то, что ссылка на файл может присутствовать одновременно в нескольких каталогах, файл может быть ссылкой на другой файл и т. д. В Linux эти возможности файловой системы весьма важны для эффективной работы, поэтому будем всюду использовать более подходящий термин "каталог".

В файловой системе, организованной при помощи каталогов, на любой файл должна быть ссылка как минимум из одного каталога, в противном случае файл просто не будет доступен внутри этой файловой системы, иначе говоря, не будет существовать.

Имена файлов и каталогов

Допустимые имена

Главные отличительные признаки файлов и каталогов -- их имена. В Linux имена файлов и каталогов могут быть длиной не более 256 символов, и могут содержать любые символы, кроме "/". Причина этого ограничения очевидна: этот символ используется как разделитель имён в составе пути, поэтому не должен встречаться в самих именах. Причём Linux всегда

различает прописные и строчные буквы в именах файлов и каталогов, поэтому "method", "Method" и "METHOD" будут тремя *разными* именами.

Есть несколько символов, допустимых в именах файлов и каталогов, которые, при этом, нужно использовать с осторожностью. Это -- так называемые **спецсимволы** "*", "\", "&", "<", ">", ";", "(", ")", "|", а также пробелы и табуляции. Дело в том, что эти символы имеют особое значение для любой командной оболочки, поэтому нужно будет специально позаботиться о том, чтобы командная оболочка воспринимала эти символы как часть имени файла или каталога. О специальном значении символа "-" для команд Linux уже шла речь на предыдущем занятии, там же обсуждалось, как изменить его интерпретацию¹. О том, зачем командной оболочке нужны спецсимволы, речь пойдёт на следующих занятиях.

Кодировки и русские имена

Как можно было заметить, пока во всех встречавшихся именах файлов и каталогов употреблялись только символы латинского алфавита и некоторые знаки препинания. Это не случайно и вызвано желанием обеспечить, чтобы приводимые примеры совершенно одинаково выглядели на любых системах. В Linux в именах файлов и каталогов допустимо использовать любые символы любого языка, однако такая свобода требует жертв, на которые Мефодий, например, пойти не смог.

Дело в том, что с давних пор каждый символ (буква) каждого языка традиционно представлялся в виде *одного* байта. Такое представление накладывает очень жёсткие ограничения на *количество* букв в алфавите: их может быть не больше 256, а за вычетом управляющих символов, цифр, знаков препинания и прочего -- и того меньше. Обширные алфавиты (например, иероглифические японский и китайский) пришлось заменять упрощённым их представлением. Вдобавок, первые 128 символов из этих 256 лучше всегда

¹ Символ "-" означает, что следующее слово -- ключ, а пробелы и табуляции разделяют параметры в командной строке

оставлять неизменными, соответствующими стандарту ASCII, включающему латиницу, цифры, знаки препинания и наиболее популярные символы из тех, что встречаются на клавиатуре печатной машинки. Интерпретация остальных 128 символов зависит от того, какая кодировка установлена в системе. Например, в русской кодировке KOI8-R 228-й символ такой таблицы соответствует букве "Д", а в западноевропейской кодировке ISO-8859-1 этот же символ соответствует букве "а" с двумя точками на ней (как у нашей буквы "ё").

Имена файлов, *записанные* на диск в одной кодировке, выглядят нелепо, если при *просмотре* каталога была установлена другая. Хуже того. Многие кодировки заполняют диапазон символов с номерами от 128 до 255 *не полностью*, поэтому соответствующего символа может вообще не быть! Это означает, что *ввести* такое искажённое имя файла с клавиатуры (например, для того, чтобы его переименовать) напрямую не удастся, придётся пускаться на разные ухищрения. Наконец, многие языки, в том числе и русский, исторически имеют *несколько* кодировок. К сожалению, в настоящее время нет стандартного способа указывать кодировку прямо в имени файла, поэтому в рамках одной файловой системы *стоит* придерживаться единой кодировки при именовании файлов.

Существует универсальная кодировка, включающая символы всех письменностей мира -- UNICODE. Стандарт UNICODE в настоящее время получает всё большее распространение и претендует на статус общего для всех текстов, хранящихся в электронной форме. *Один* символ в UNICODE может занимать *больше* одного байта -- и в этом главный его недостаток, так как множество полезных прикладных программ, отлично работающих с *однобайтными* кодировками, необходимо основательно или даже полностью перерабатывать для того, чтобы научить их обращаться с UNICODE.

Это не означает, что называя файлы, не следует использовать языки, отличные от английского. Пока точно известно, в какой кодировке задано имя файла -- проблем не возникнет. Однако Мефодий решил, что гарантий в

передаче названного по-русски файла на какую-нибудь *другую* систему можно добиться только передавая вместе с ним настройку кодировки, даже две: в своей системе и в системе адресата (неизвестно какой!). Другой, гораздо более лёгкий, способ передать файл -- использовать в его названии *только* символы ASCII.

Расширения

Многим пользователям знакомо понятие **расширение** -- часть имени файла после точки, обычно ограничивающаяся несколькими символами и указывающая на тип содержащихся в файле данных. В файловой системе Linux нет никаких предписаний по поводу расширения: в имени файла может быть любое количество точек (в том числе и ни одной), а после последней точки может быть любое количество символов². Хотя расширения не обязательны и не навязываются технологией в Linux, они широко используются: расширение позволяет человеку или программе, не открывая файл, только по его имени определить, какого типа данные в нём содержатся. Однако нужно учитывать, что расширение -- это только набор соглашений по наименованию файлов разных типов. Строго говоря, данные в файле могут не соответствовать заявленному расширению по той или иной причине, поэтому всецело полагаться на расширение просто нельзя.

Определить тип содержимого файла можно и на основании самих данных. Многие форматы предусматривают указание в начале файла, как следует интерпретировать дальнейшую информацию: как программу, исходные данные для текстового редактора, страницу HTML, звуковой файл, изображение или что-то другое. В распоряжении пользователя Linux всегда есть утилита `file`, которая предназначена именно для определения типа данных, содержащихся в файле.

² В отличие от старых файловых систем, организованных по принципу "8+3" (DOS, ISO9660 и т. п.), где в имени файла допустимо не более одной точки и расширение может быть не длиннее 3-х символов. Это ограничение определило вид многих из известных сегодня расширений файлов, например, ".txt" для текстового файла.

```
[methody@localhost methody]$ file -- -filename-with-  
-filename-with-: ASCII English text  
[methody@localhost methody]$ file /home/methody  
/home/methody: directory
```

Определение типа данных в файле

Мефодий, забыв, что содержится в файле "-filename-with-", который он создал на прошлой лекции, хотел было уже посмотреть его содержимое при помощи команды `cat`. Однако его остановил Гуревич, который посоветовал сначала выяснить, что за данные содержатся в этом файле. Не исключено, что это двоичный файл исполняемой программы, в таком файле могут встречаться последовательности, которые случайно совпадут с управляющими последовательностями терминала. Поведение терминала после этого может стать непредсказуемым, а неопытный Мефодий вряд ли сможет с ним справиться. Мефодий получил вполне точный ответ от утилиты `file`: в его файле -- английский текст в кодировке ASCII. `file` умеет различать очень многие типы данных и почти наверняка выдаст правильную информацию. Эта утилита никогда не "доверяет" расширению файла (если оно присутствует), и анализирует сами данные. `file` различает не только разные данные, но и разные типы файлов, в частности, сообщит, если исследуемый не является обычным файлом, а, например, каталогом.

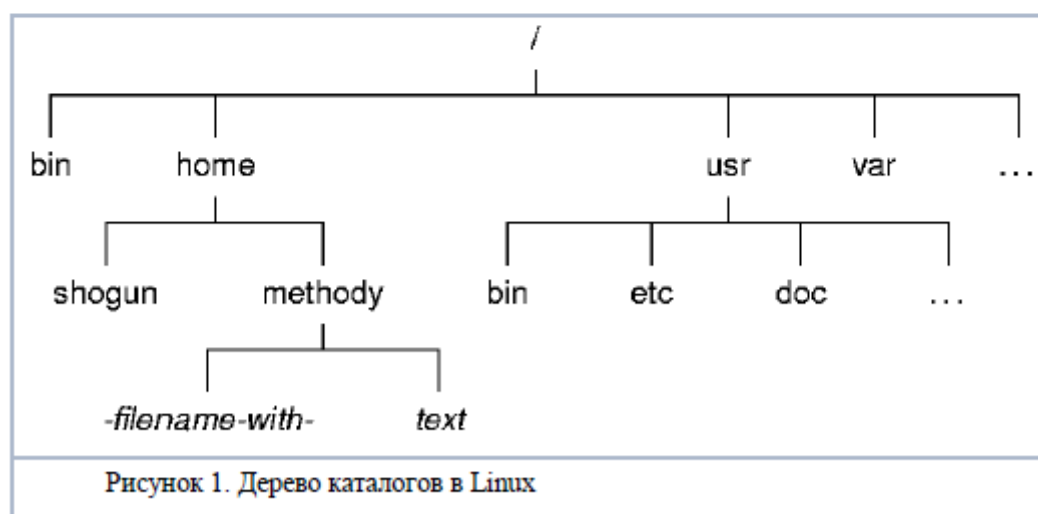
Дерево каталогов

Понятие каталога позволяет *систематизировать* все объекты, размещённые на носителе данных (например, на диске). В большинстве современных файловых систем используется иерархическая модель организации данных: существует один каталог, объединяющий все данные в файловой системе -- это "корень" всей файловой системы, **корневой каталог**. Корневой каталог может содержать любые объекты файловой системы, и в частности, подкаталоги (каталоги первого уровня вложенности). Те, в свою очередь, также могут содержать любые объекты файловой системы и подкаталоги (второго уровня вложенности) и т. д. Таким образом, *всё*, что

записано на диске -- файлы, каталоги и специальные файлы -- обязательно "принадлежит" корневому каталогу: либо непосредственно (содержится в нём), либо на некотором уровне вложенности.

Иерархию вложенных друг в друга каталогов можно соотнести с иерархией данных в системе: объединить тематически связанные файлы в каталог, тематически связанные каталоги -- в один общий каталог и т. д. Если строго следовать иерархическому принципу, то чем глубже будет **уровень вложенности** каталога, тем более частным признаком должны быть объединены содержащиеся в нём данные. Если этому принципу *не* следовать, то вскоре окажется гораздо проще складывать *все* файлы в один каталог и искать нужный среди них, чем проделывать такой поиск по всем подкаталогам системы. Однако в этом случае о какой бы то ни было *систематизации* файлов говорить не приходится.

Структуру файловой системы можно представить наглядно в виде дерева³, "корнем" которого является корневой каталог, а в вершинах расположены все остальные каталоги. На рисунке 1 изображено дерево каталогов, курсивом обозначены имена файлов, прямым начертанием -- имена каталогов.



³ Здесь имеется в виду дерево в строгом математическом смысле: ориентированный граф без циклов с одной корневой вершиной, в котором в каждую вершину входит ровно одно ребро.

В любой файловой системе Linux всегда есть только один **корневой каталог**, который называется `/`. Пользователь Linux всегда работает с единым деревом каталогов, даже если разные данные расположены на разных носителях: нескольких жёстких или сетевых дисках, съёмных дисках, CD-ROM и т. п.⁴ Для того, чтобы подключать и отключать файловые системы на разных устройствах в одно общее дерево, используются процедуры **монтирования** и **размонтирования**, о которых речь пойдёт на следующих занятиях. После того, как файловые системы на разных носителях подключены к общему дереву, содержащиеся на них данные доступны так, как если бы все они составляли единую файловую систему: пользователь может даже не знать, на каком устройстве какие файлы хранятся.

Положение любого каталога в дереве каталогов точно и однозначно описывается при помощи **полного пути**. Полный путь всегда начинается от корневого каталога и состоит из перечисления всех вершин, встретившихся при движении по рёбрам дерева до искомого каталога включительно. Названия соседних вершин разделяются символом `/` ("слэш"). В Linux полный путь, например, до каталога `methody` в файловой системе, приведённой на рисунке записывается следующим образом: сначала символ `/`, обозначающий корневой каталог, затем к нему добавляется `home`, затем разделитель `/`, за которым следует название искомого каталога `methody`, в результате получается полный путь `/home/methody`⁵.

Расположение файла в файловой системе аналогичным образом определяется при помощи полного пути, только последним элементом в данном случае будет не название каталога, а название файла. Например,

⁴ Это отличается от технологии, применяемой в Windows или Amiga, где для каждого устройства, на котором есть файловая система, используется свой корневой каталог, обозначенный литерой, например `a`, `c`, `d` и т. д.

⁵ Весьма похожий способ записи полного пути используется в системах Windows, с той разницей, что корневой раздел обозначается литерой устройства с последующим двоеточием, а в качестве разделителя используется символ `\` ("обратный слэш").

полный путь до созданного Мефодием файла "-filename-with-" будет выглядеть так: "/home/methody/-filename-with-"⁶.

Организация каталогов файловой системы в виде дерева не допускает появления циклов: т. е. каталог не может содержать в себе каталог, в котором содержится сам. Благодаря этому ограничению полный путь до любого каталога или файла в файловой системе всегда будет *конечным*.

Размещение компонентов системы: Стандарт FHS

Попробуем более подробно разобраться, как устроено дерево каталогов Linux и где что в нём можно найти. Фрагмент дерева каталогов типичной файловой системы Linux (*Some Linux*, которую использует Мефодий) приведён на рисунке 1. Мефодий решил обследовать свою файловую систему, начиная с **корневого каталога**: Гуревич посоветовал использовать для этого команду `ls каталог`, где *каталог* -- это полный путь к каталогу: утилита `ls` выведет список всего, что в этом каталоге содержится.

```
[methody@localhost methody]$ ls /  
bin  dev  home  mnt  root  tmp  var  
boot etc  lib   proc /sbin  usr  
[methody@localhost methody]$
```

Стандартные каталоги в /

Утилита `ls` вывела список подкаталогов корневого каталога. Этот список будет таким же или почти таким же в любом дистрибутиве Linux. В корневом каталоге Linux-системы обычно находятся только подкаталоги со *стандартными* именами. Более того, не только имена, но и *тип данных*, которые могут попасть в тот или иной каталог, также регламентированы этим стандартом. Этот стандарт называется **Filesystem Hierarchy Standard** ("стандартная структура файловых систем").

⁶ Полный путь к каталогу формально ничем не отличается от пути к файлу, т. е. по полному пути нельзя сказать наверняка, является ли его последний элемент файлом или каталогом. Чтобы отличать путь к каталогу, иногда используют запись с символом "/" в конце пути: "/home/methody/".

Опишем кратко, что находится в каждом из подкаталогов корневого каталога. Мы не будем приводить полные списки файлов для каждого описываемого каталога, а Мефодий сможет просмотреть их при помощи команды `ls имя каталога`.

`/bin`

- Название этого каталога происходит от слова "binaries" ("двоичные", "исполняемые"). В этом каталоге находятся исполняемые файлы самых необходимых утилит. Сюда попадают такие программы, которые могут понадобиться системному администратору или другим пользователям для устранения неполадок в системе или при восстановлении после сбоя.

`/boot`

- "Boot" -- загрузка системы. В этом каталоге находятся файлы, необходимые для самого первого этапа загрузки: загрузки ядра и, обычно, само ядро. Пользователю практически никогда не требуется непосредственно работать с этими файлами.

`/dev`

- В этом каталоге находятся все имеющиеся в системе файлы-дырки: файлы особого типа, предназначенные для обращения к различным системным ресурсам и устройствам (англ. "devices" -- "устройство", отсюда и сокращённое название каталога). Например, файлы `/dev/ttyN` соответствуют виртуальным консолям, где *N* -- номер виртуальной консоли. Данные, введённые пользователем на первой виртуальной консоли, система считывает из файла `/dev/tty1`, в этот же файл записываются данные, которые нужно вывести пользователю на эту консоль. В файлах-дырках в действительности не хранятся никакие данные, при их помощи данные *передаются*.

`/etc`

- Каталог для системных конфигурационных файлов. Здесь хранится информация о специфических настройках данной системы: информация о зарегистрированных пользователях, доступных ресурсах, настройках различных программ.

`/home`

- Здесь расположены каталоги, принадлежащие пользователям системы -- домашние каталоги, отсюда и название "home". Отделение всех файлов, создаваемых пользователями, от прочих системных файлов даёт очевидное преимущество: серьёзное повреждение системы или необходимость обновления не затронет наиболее ценной информации -- пользовательских файлов.

`/lib`

- Название этого каталога -- сокращение от "libraries" (англ. "библиотеки"). Библиотеки -- это собрания наиболее стандартных функций, необходимых многим программам: операций ввода/вывода, рисования элементов графического интерфейса и проч. Чтобы не включать эти функции в текст каждой программы, используются стандартные функции библиотек -- это значительно экономит место на диске и упрощает написание программ. В этом каталоге содержатся библиотеки, необходимые для работы наиболее важных системных утилит (размещённых в `/bin` и `/sbin`).

`/mnt`

- Каталог для монтирования (от англ. "mount") -- временного подключения файловых систем, например, на съёмных носителях (CD-ROM и др.).

`/proc`

- В этом каталоге все файлы "виртуальные" -- они располагаются не на диске, а в оперативной памяти. В этих файлах содержится информация о программах (процессах), выполняемых в данный момент в системе.

`/root`

- Домашний каталог администратора системы -- пользователя `root`. Смысл размещать его отдельно от домашних каталогов остальных пользователей состоит в том, что `/home` может располагаться на отдельном устройстве, которое не всегда доступно (например, на сетевом диске), а домашний каталог `root` должен присутствовать в любой ситуации.

`/sbin`

- Каталог для важнейших системных утилит (название каталога -- сокращение от "system binaries"): в дополнение к утилитам `/bin` здесь находятся программы, необходимые для загрузки, резервного копирования, восстановления системы. Полномочия на исполнение этих программ есть только у системного администратора.

`/tmp`

- Этот каталог предназначен для временных файлов: в таких файлах программы хранят промежуточные данные, необходимые для работы. После завершения работы программы временные файлы теряют смысл и должны быть удалены. Обычно каталог `/tmp` очищается при каждой загрузке системы.

`/usr`

- Каталог `/usr` -- это "государство в государстве". Здесь можно найти такие же подкаталоги `bin`, `etc`, `lib`, `sbin`, как и в корневом каталоге. Однако в корневой каталог попадают только утилиты, *необходимые* для загрузки и восстановления системы в аварийной ситуации, *все остальные* программы и данные располагаются в подкаталогах `/usr`. Прикладных программ в современных системах обычно установлено очень много, поэтому этот раздел файловой системы может быть очень большим.

`/var`

- Название этого каталога -- сокращение от "variable" ("переменные" данные). Здесь размещаются те данные, которые создаются в процессе работы разными программами и предназначены для передачи другим программам и системам (очереди печати и электронной почты и др.) или для сведения системного администратора (системные журналы, содержащие протоколы работы системы). В отличие от каталога `/tmp` сюда попадают те данные, которые могут понадобиться после того, как создавшая их программа завершила работу.

Стандарт FHS регламентирует не только перечисленные каталоги, но и их подкаталоги, а иногда даже приводит список конкретных файлов, которые должны присутствовать в определённых каталогах⁷. Этот стандарт последовательно соблюдается во всех Linux-системах, хотя и не без горячих споров между разработчиками при выходе каждой новой его версии.

Стандартное размещение файлов позволяет и человеку, и даже программе предсказать, где находится тот или иной компонент системы. Для человека это означает, что он сможет быстро сориентироваться в любой системе Linux (где файловая система организована в соответствии со стандартом) и найти то, что ему нужно. Для программ стандартное расположение файлов -- это возможность организации автоматического взаимодействия между разными компонентами системы.

Мефодий уже успел воспользоваться некоторыми преимуществами, которые даёт использование стандартного расположения файлов: на предыдущих занятиях он запускал утилиты, не указывая полный путь к исполняемому файлу, например, `cat` вместо `/bin/cat`. Командная оболочка "знает", что исполняемые файлы располагаются в каталогах `/bin`, `/usr/bin` и т. д. -- именно в этих каталогах она ищет исполняемый файл `cat`. Благодаря этому

⁷ Краткое описание стандартной иерархии каталогов Linux можно получить, отдав команду `man hier`. Полный текст и последнюю редакцию стандарта FHS можно прочесть по адресу <http://www.pathname.com/fhs/>.

каждая вновь установленная в системе программа немедленно оказывается доступна пользователю из командной строки, для этого не требуется ни перезагружать систему, ни запускать никаких процедур -- достаточно просто поместить исполняемый файл в один из соответствующих каталогов.

Рекомендации стандарта по размещению файлов и каталогов основываются на принципе разносить в разные подкаталоги файлы, которые по-разному используются в системе. По типу использования файлов их можно разделить на следующие группы:

пользовательские/системные файлы

- Пользовательские файлы -- это все файлы, созданные пользователем и не принадлежащие ни одному из компонентов системы. О пользе разграничения пользовательских и системных файлов речь уже шла выше.

изменяющиеся/неизменные файлы

- К неизменным файлам относятся все статические компоненты программного обеспечения: библиотеки, исполняемые файлы и др. -- всё, что не изменяется само без вмешательства системного администратора. Изменяющиеся -- это те, которые изменяются без вмешательства человека в процессе работы системы: системные журналы, очереди печати и пр. Выделение неизменных файлов в отдельную структуру (например, /usr) позволяет использовать соответствующую часть файловой системы в режиме "только чтение", что уменьшает вероятность случайного повреждения данных и позволяет использовать для хранения этой части файловой системы CD-ROM и другие носители, доступные только для чтения.

разделяемые/неразделяемые файлы

- Это разграничение становится полезным, если речь идёт о сети, в которой работает несколько компьютеров. Значительная часть информации при этом может храниться на одном из компьютеров

и использоваться всеми остальными по сети (к такой информации относятся, например, многие программы и домашние каталоги пользователей). Однако часть файлов нельзя разделять между системами (например, файлы для начальной загрузки системы).

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Определение Файла
2. Определение Файловой системы
3. Определение Каталога
4. Требования к допустимым именам файлов и каталогов
5. Назначение Расширения файлов. Необходимо ли использовать расширения в Linux?
6. Назначение утилиты `file`.
7. Структура файловой системы в Linux. Понятие корневого каталога.
8. Есть ли в Linux диск C?
9. Для чего предназначены процедуры монтирования/размонтирования?
10. Пояснить назначение символа «/» в полном пути к файлу `"/home/methody"`.
11. Назначение утилиты `ls`.
12. Назначение каталогов: `/bin`, `/boot`, `/dev`, `/etc`, `/home`, `/lib`, `/mnt`, `/proc`, `/root`, `/sbin`, `/tmp`, `/usr`, `/var`.
13. Рекомендации стандарта FHS по размещению файлов и каталогов.