**Системное администрирование Linux.**

Целью занятия является формирование и закрепление навыков по системному администрированию Linux систем для архитектуры х86.

Задачи:

1. Овладеть базовым функционалом системного администрирования Linux-систем.

2. Научиться определять и устанавливать значения необходимых параметров при использовании команд оболочки Shell.

**Необходимые теоретические сведения.**

**Файлы в ОС Linux**

Имя файла может содержать любые буквы, символы подчеркивания и цифры. Мож­но включать в имена точки и запятые. При этом имя файла не должно начинаться ни с цифры, ни с точки (за исключением некоторых особых случаев). Остальные символы в частности символ косой черты, вопросительные знаки и звездочки, зарезервированы для применения в системе в качестве специальных символов и в именах файлов использоваться не должны. Максимальная длина имени файла — 256 символов.

Строгим является только одно ограничение — на длину имени файла.

Имя файла может включать в себя расширение. Для отделения расширения от самого имени служит точка. Расширения применяются для разбивки файлов по категориям.

Все файлы в ОС Linux имеют один и тот же формат - байтовый поток. *Байтовый поток* представляет собой просто последовательность байтов. Это позволяет распространять понятие файла в системе Linux на все компоненты данных. Каталоги и устройства­ классифицируются как файлы. Рассматривая все эти объекты как файлы, Linux позволяет упростить организацию данных и обмен ими. Данные, записанные в файле, можно посылать непосредственно на устройство, например на экран, потому что устройство сопрягается с системой, используя тот же формат байтового потока, который применяется в обычных файлах.

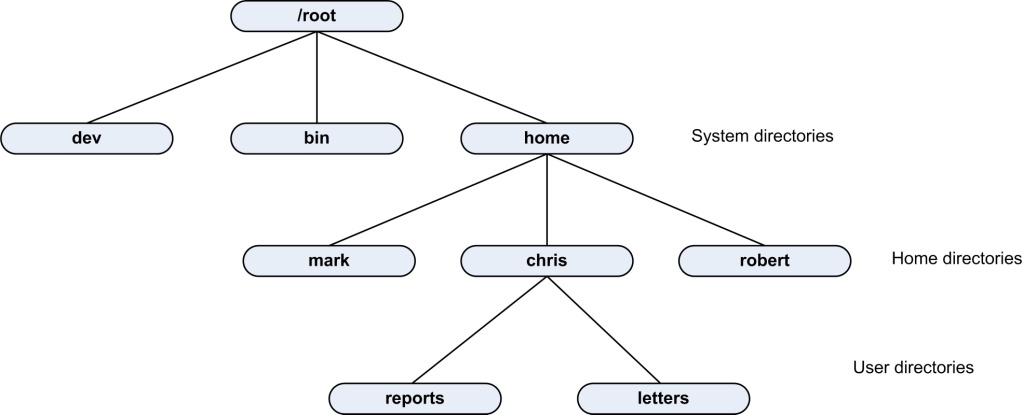
Этот же формат файлов используется при создании других компонентов операци­онной системы. Интерфейс, обеспечивающий доступ к какому-либо устройству, например к экрану или клавиатуре, проектируется как файл. Другие компоненты, в частности каталоги, сами по себе являются файлами типа потоков байтов, но имеют особую внут­реннюю организацию. Файл каталога содержит информацию о каталоге, оформленную в специальном формате каталога. Поскольку все эти различные компоненты рассмат­риваются как файлы, можно сказать, что они представляют собой различные *типы файлов.* Устройство посимвольного ввода-вывода — один тип файла. Каталог — другой тип файла. Число типов файлов зависит от конкретной реализации ОС Linux, однако существуют пять стандартных типов: обычные файлы, файлы каталогов, каналы, дей­ствующие по принципу очереди, файлы устройств посимвольного ввода-вывода и файлы устройств поблочного ввода-вывода.

**Файловая структура**

Файлы в операционной системе Linux организованы в иерархическую систему ка­талогов. Каталог может содержать файлы и другие каталоги. В этом смысле каталоги выполняют две важные функции. Во-первых, в каталоге хранятся файлы, подобно пап­кам в ящике картотеки, а во-вторых, каталог соединяется с другими каталогами, как ветвь дерева соединяется с другими ветвями. По отношению к файлам каталоги вы­полняют роль ящиков картотеки, в каждом из которых хранятся несколько папок. Для того чтобы взять одну из них, нужно открыть ящик. Следует отметить, однако, что, в отличие от ящиков картотеки, каталоги могут содержать не только файлы, но и другие каталоги. Именно таким образом каталог может соединяться с другим каталогом.

Из-за сходства с деревом такую структуру часто называют *древовидной структурой,* данную структуру можно назвать *структурой «с родительско-дочерними связями».* Аналогичным образом любой каталог является подкаталогом другого каталога. Каждый каталог может содержать множество подкаталогов, но сам должен быть потомком только одного родительского каталога.

Файловая структура ОС Linux разветвляется на несколько каталогов, начиная с кор­невого, /. В корневом каталоге имеется несколько системных каталогов, которые вклю­чают файлы и программы, относящиеся к самой ОС Linux. Корневой каталог, кроме того, содержит каталог home, в котором могут храниться начальные каталоги всех поль­зователей системы. Начальный каталог каждого пользователя, в свою очередь, будет включать в себя каталоги, которые пользователь создает для своих нужд. Каждый из этих каталогов тоже может содержать каталоги. Все эти вложенные каталоги ответвля­ются от начального каталога пользователя, как показано на рис. 1.



**Рис. 1.** Файловая структура Linux начинается **с** корневого каталога

**Начальные каталоги**

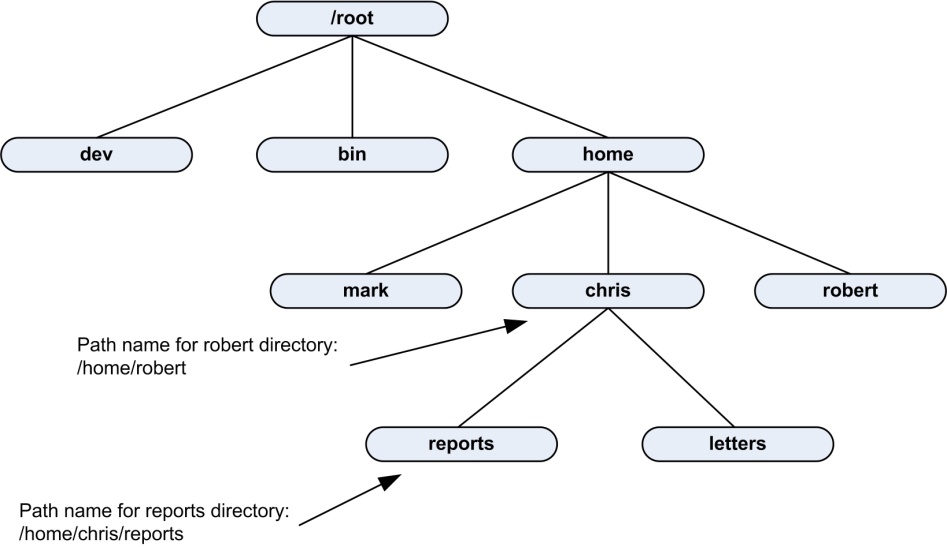
Зарегистрировавшись в системе, вы попадаете в свой начальный каталог. Имя, при­своенное этому каталогу системой, совпадает с вашим регистрационным именем. Все файлы, создаваемые для нового пользователя, помещаются в его начальный каталог. В этом каталоге можно создавать подкаталоги и размещать в них файлы. Создавать свои начальные каталоги могут и другие пользователи системы. Имя начального ката­лога каждого пользователя совпадает с регистрационным именем этого пользователя.

Если при проведении какой-либо операции над файлами имя каталога не указывается, то используется каталог по умолчанию, который называют также *рабочим каталогом.*

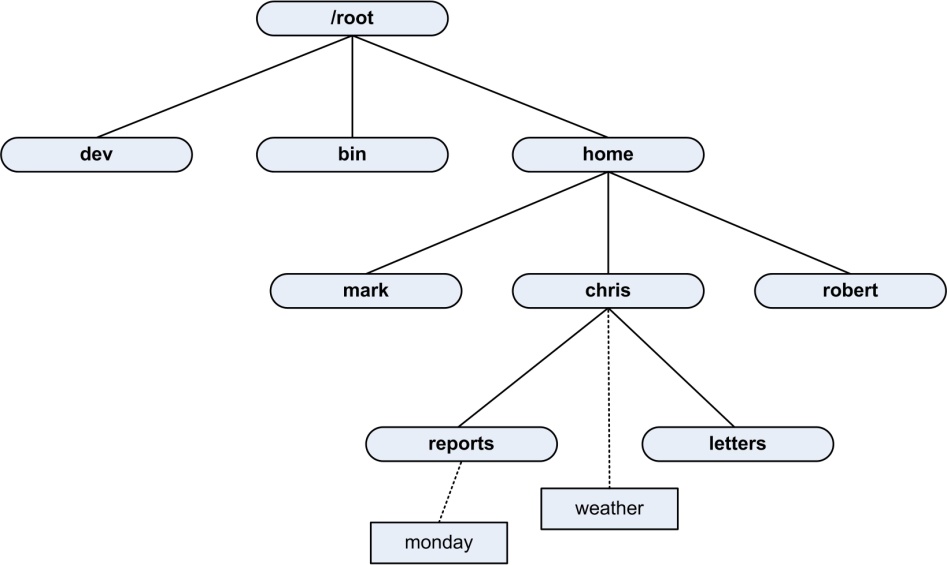
При регистрации в системе в качестве рабочего принимается ваш начальный каталог, имя которого обычно совпадает с вашим регистрационным именем. Рабочий каталог можно сменить с помощью команды cd. В процессе смены рабочего каталога вы пе­реходите из одного каталога в другой.

**Путевые имена**

Имя, которое дается каталогу или файлу при его создании, не является полным. Полным именем каталога является его *путевое имя.* Иерархические связи, существую­щие между каталогами, образуют пути, и эти пути можно использовать для однозначного указания каталога или файла и обращения к нему. На рис. 2 показан путь от кор­невого каталога, /, через каталог home в каталог robert, и путь от корневого каталога через каталоги home и chris в каталог reports. Путевое имя файла weather (Рис. 3) состоит из имен корневого каталога, каталогов home и chris и имени файла: /home/chris/weather (корневой каталог обозначен первой косой чертой).



**Рис. 2.** Путевые имена каталогов



**Рис. 3.** Путевое имя для файла weather: /home/chris/weather

Путевые имена могут быть полными и сокращенными. *Полное путевое имя* — это полное имя файла или каталога, начинающееся символом корневого каталога. *Сокра­щенное путевое имя* начинается символом рабочего каталога и представляет собой обо­значение пути к файлу относительно вашего рабочего каталога. В структуре каталогов, изображенной на рис. 3, chris — рабочий каталог, а сокращенное путевое имя файла monday — reports/monday. Полное путевое имя этого файла — /home/chris/re-ports/monday.

**Системные каталоги**

Корневой каталог, являющийся началом файловой структуры ОС Linux, содержит ряд системных каталогов. В них хранятся файлы и программы, служащие для управления системой и ее сопровождения. Многие из этих каталогов включают подкаталоги с про­граммами, предназначенными для выполнения конкретных задач. Основные системные каталоги перечислены в табл. 1.

**Таблица 1.** Стандартные системные каталоги в ОС Linux

|  |  |
| --- | --- |
| **/** | Служит начальной единицей файловой системы и называется *корнгеым* |
| **/home** | Содержит *начальные* каталоги пользователей |
| **/bin** | Содержит все стандартные команды и утилиты |
| **/usr** | Содержит файлы и команды, используемые системой; разбит на несколько подкаталогов |
| **/usr/bim** | Содержит команды и утилиты, применяемые пользователем |
| **/usr/sbin** | Содержит команды системного администрирования |
| **/usr/lib** | Содержит библиотеки языков программирования |
| **/usr/doc** | Содержит документацию ОС Linux |
| **/usr/man** | Содержит файлы справочных руководств |
| **/usr/spool** | Содержит буферные файлы (например, создаваемые для заданий по сети) |
| **/sbin** | Содержит команды администрирования системы для начальной загрузки |
| **/var** | Содержит изменяющиеся файлы, например файлы почтовых ящиков |
| **/dev** | Содержит файлы, обеспечивающие взаимодействие пользователя с устройствами на пример с терминалами и принтерами |
| **/etc** | Содержит файлы конфигурации системы и прочие системные файлы |

**Управление каталогами: mkdir, rmdir, ls, cd и pwd**

Пользователь может создавать и удалять собственные каталоги, а также изменять свой рабочий каталог. Команды, используемые для управления каталогами, приведены в табл. 2.

**Таблица 2.** Команды управления каталогами

|  |  |
| --- | --- |
| **mkdir** | Создает каталог: **mkdir reports** |
| **rmdir** | Удаляет каталог: **rmdir letters** |
| **ls -F** | Выводит список имен файлов и **каталогов,** ставя после имени ка­талога косую черту: **ls** -F **today reports/ letters/** |
| **ls -R** | Выводит содержимое рабочего каталога со всеми его подкаталогами |
| **cd** | Переход в указанный каталог, который становится рабочим ката­логом; **cd** без имени каталога означает возврат в начальный ката­лог:  **cd reports**  **cd** |
| **pwd** | Сообщает путевое имя рабочего каталога:  **pwd /home/chris/reports** |
| **Имя\_каталога/имя\_файла** | Косая черта используется в путевых именах для разделения имен каталогов. Если речь идет о путевых именах файлов, то косая черта отделяет имена каталогов от имени файла:  **cd /home/chris/reports**  **cat /home/chris/reports/mydata** |
| **..** | Обозначает родительский каталог. Этот символ можно использовать как параметр или часть путевого имени:  **cd ..**  **mv ../larisa oldletters** |
| **.** | Обозначает рабочий каталог. Этот символ можно использовать как параметр или как часть путевого имени  **ls .**  **mv ../aleina .** |
| **~** | Тильда — это специальный символ, который обозначает путевое имя начального каталога. Данное средство полезно в том случае, если нужно использовать полное путевое имя файла или каталога:  **ср monday ~/today**  **mv tuesday -/weather** |

**Перемещение и копирование каталогов**

Система Linux позволяет копировать и перемещать целые каталоги. В качестве пер­вого параметра команды ср и mv могут использовать имя каталога, позволяя копировать и перемещать подкаталоги из одного каталога в другой (табл. 3). Первый параметр — имя перемещаемого или копируемого каталога, а второй — имя каталога, в который он будет помещен. При перемещении и копировании каталогов действует та же струк­тура путевых имен, что и при соответствующих операциях с файлами.

**Таблица 3.** Операции с файлами и каталогами

|  |  |
| --- | --- |
| cp имя\_файла имя\_файла | Копирует файл. Для команды ср обязательны два парамет­ра: исходный файл и имя копии. При копировании из од­ного каталога в другой можно использовать путевые имена файлов  ср today reports/Monday |
| cp -r имя\_каталога имя\_каталога | Копирует подкаталог из одного каталога в другой. Скопи­рованный каталог содержит все свои подкаталоги  ср -г letters/thankyou oldletters |
| mv имя\_файла имя\_файла | Перемещает (переименовывает) файл. Команда mv исполь­зуется с двумя параметрами: имя перемещаемого файла и новое имя файла либо путевое имя каталога. Если это имя каталога, то файл перемещается в данный каталог и его пу­тевое имя изменяется  mv today /home/chris/reports |
| mv имя\_каталога имя\_каталога | Перемещает каталоги. В данном случае первый и последний параметры — каталоги  mv letters/thankyou oldletters |
| rm список\_файлов | Удаляет файл. Использует в качестве параметров любое количество имен файлов. Удаляет ссылки на файл.  rm today weather weekend |

Хотя все [файлы в системе Linux](#bookmark13) логически соединены в одно общее дерево, сами файлы размещаются на различных запоминающих устройствах, например на жестких дисках и компакт-дисках. Файлы, записанные на запоминающих устройствах, органи­зованы в файловые системы. Дерево каталогов в ОС Linux может охватывать несколько файловых систем, каждая из которых размещена на отдельном устройстве. Сами файлы организованы в единое файловое дерево, вершиной которого является корневой каталог.

Файлы той или иной файловой системы остаются отделенными от дерева каталогов до тех пор, пока вы явно не присоедините их к этому дереву. В каждой файловой системе файлы организованы в отдельное дерево каталогов. Это дерево можно рас­сматривать как поддерево, которое необходимо присоединить к основному дереву ка­талогов. Например, на дискете с файлами ОС Linux — свое дерево каталогов. Это поддерево нужно присоединить к основному дереву, расположенному в разделе жесткого диска. Пока это не сделано, доступа к файлам на дискете у вас не будет.

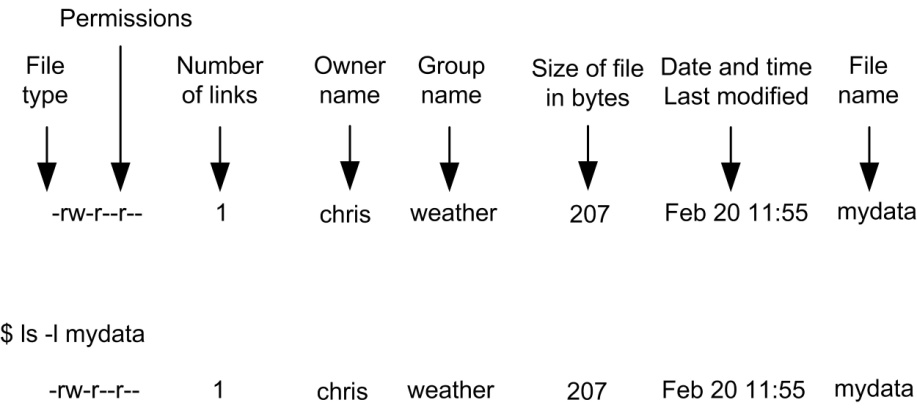
Присоединение файловой системы, расположенной на запоминающем устройстве, к основному дереву каталогов называется *монтированием устройства.* Операция монтирования предполагает присоединение де­рева каталогов, находящегося на устройстве памяти, к указанному пользователем ка­талогу. Лишь после этого можно перейти в присоединенный каталог и обращаться к его файлам. Каталог, к которому вы хотите присоединить файлы, находящиеся на за­поминающем устройстве, называется *точкой монтирования.*

Монтирование файловой системы может осуществляться только привилегированным пользователем, поскольку эта задача относится к функциям системного администратора. После монтирования компакт-диска его невозможно будет свободно извлечь и вставить новый (также не следует вынимать из дисковода смонти­рованную дискету). Так, привод компакт-диска останется запертым до тех пор, пока вы не демонтируете компакт-диск. После этого диск можно извлечь и вставить другой, который, прежде чем использовать, необходимо смонтировать. При каждой замене ком­пакт-диска или дискеты вам необходимо будет повторять операцию монтирования - демонтирования.

**Практическая часть**

**Отображение информации о файле: команда ls**

Как показано на рис. 4, команда ls -l отображает подробную информацию о файле. Вначале отображаются права доступа, за ними следуют ссылки, имя владельца файла, имя группы, в которую входит пользователь, размер файла в байтах, дата и время последнего обновления файла и его имя. Права доступа указывают, кто может иметь доступ к файлу: пользователи, члены группы или все другие пользователи. Из рис. 4 видно, что файл mydataотносится к числу обычных файлов. Количество ссылок равно 1; это значит, что файл не имеет других имен и на него нет других ссылок. Имя владельца chrisбовпадает с регистра­ционным именем, а имя группы будет weather**.** Возможно, в группу weatherвходят и другие пользователи. Размер файла — 207 байтов. Файл обновлялся последний раз 20-го февраля, в 11:55. Имя файла — mydata.

****

**Рис. 4.** Информация о файле, отображаемая при использовании команды ls с опцией -l

**Файловая структура**

Путевые имена могут быть полными и сокращенными. *Полное путевое имя* — это полное имя файла или каталога, начинающееся символом корневого каталога. *Сокра­щенное путевое имя* начинается символом рабочего каталога и представляет собой обо­значение пути к файлу относительно вашего рабочего каталога.

Для упрощения доступа к своим файлам пользователь может указывать специальный символ «тильда» (~), который представляет полное путевое имя начального каталога. В сле­дующем примере пользователь, работая в каталоге thankyou, обращается к файлу weather в своем начальном каталоге, поместив перед именем файла weather символы «тильда» и «косая черта»:

**pwd**

/home/chris/letters/thankyou

**cat** ~/weather

raining and warm

**Отображение файлов: cat и more**

Во многих случаях бывает необходимо просматривать содержимое файла. Команды cat и moreвыводят содержимое файла на экран. Название команды catобразовано путем сокращения слова *concatenate.* В приведенном ниже примере осуществляется вывод текста файла на экран:

**cat** mydata

computers

Команда catвыводит на экран сразу весь текст файла. Если файл большой, то текст очень быстро мелькает на экране. Для устранения этого недостатка служит ко­манда more,с помощью которой текст на экран можно выводить порциями. Эта ко­манда вызывается с именем файла, который вы хотите просмотреть:

**more** mydata

Когда moreвызывает файл, отображает его первый фрагмент, умещающийся на экране. Для отображения следующего фрагмента нажимается клавиша [f] или клавиша пробела. Для возврата к предыдущему тексту используется клавиша [b]. Нажав клавишу [q], можно в любой момент выйти из данной программы.

**Управление каталогами: mkdir, rmdir, ls, cd и pwd**

Пользователь может создавать и удалять собственные каталоги, а также изменять свой рабочий каталог. Для этого служат команды mkdir, rmdir и cd.

Каталоги создаются и удаляются, соответственно, командой mkdir и командой rmdir.В том и другом случае можно использовать путевые имена каталогов. В следующем примере пользователь сначала создает каталог reports, а затем, используя полное путевое имя, — каталог letters.

**mkdir** reports

**mkdir** /home/chris/letters

Для удаления каталога нужно применить команду rmdir с именем этого каталога. В приведенном ниже примере пользователь сначала удаляет командой rmdir каталог reports, а затем, указав полное путевое имя, — каталог letters.

**rmdir** reports

**rmdir** /home/chris/letters

В каждом каталоге можно создавать другие каталоги, осуществляя, по сути дела вложение одного каталога в другой. Команда cd позволяет переходить из одного ката­лога в другой, однако никакого указателя на то, в каком каталоге вы в данный момент находитесь, нет. Для того чтобы определить, в какой каталог вы перешли, выполните команду pwd, которая сообщит полное путевое имя рабочего каталога, как показано в следующем примере. Путевое имя состоит из имен рабочего каталога dylan и каталога, частью которого он является, home. Имена каталогов разделены косой чертой. Корневой каталог обозначен первой косой чертой.

**pwd**

/home/dylan

Файловые команды, например ls и ср, будут манипулировать файлами, находящи­мися в рабочем каталоге, если иного не указано.

В качестве параметра команда cd использует имя каталога, в который вы хотите перейти. В качестве имени каталога может быть указано имя подкаталога рабочего каталога или полное путевое имя любого каталога в системе. Для перехода в начальный каталог достаточно ввести команду cd без параметров.

**pwd**

/home/dylan

**cd** props

**pwd**

/home/dylan/props

**cd** /home/chris/letters

**pwd**

/home/chris/letters

Для обозначения родительского каталога можно пользоваться двумя точками (..). Этот специальный символ обозначает путевое имя родительского каталога. Его допус­кается использовать в команде cd для перехода обратно в родительский каталог, таким образом вновь делая этот каталог текущим. В следующем примере пользователь пере­ходит в каталог letters, а затем возвращается в начальный каталог.

**cd** letters

**pwd**

/home/chris/letters

**cd ..**

**pwd**

/home/chris

Используя команду cd с символом .., можно возвращаться из каталога нижнего уровня, последовательно переходя в родительские каталоги по дереву каталогов.

**Операции с файлами и каталогами, монтирование**

В ОС Linux предусмотрен набор команд, которые обеспечивают поиск, копиро­вание, переименование и удаление файлов. При наличии в системе большого числа файлов иногда приходится также выполнять поиск конкретного файла. Команды представляют собой сокращенную форму слов, состоящую из двух символов. Команда ср обозначает «сору» и позволяет копировать файл, mv — «move» и дает возможность перемещать либо переименовывать файл, rm — «remove», она приводит к удалению файла.

**Перемещение и копирование файлов**

В следующем примере пользователь копирует файл proposalв новый файл oldprop**:**

**cp** proposal oldprop

Может случиться так, что при копировании файла с помощью команды ср вы непреднамеренно уничтожите другой файл. При создании копии посредством этой ко­манды сначала создается файл, а затем в него копируются данные. Если какой-нибудь файл уже имеет то же имя, которое вы указали для выходного файла, первый из них уничтожается и создаётся новый файл с этим именем. В некотором смысле можно сказать, что файл оригинала перекрывается новой копией. В следующем примере файл proposal перекрывается файлом newprop. Файл proposal уже существует.

**cp** newprop proposal

В большинстве дистрибутивов Linux существует возможность настроить систему для выявления подобных случаев. Если же такой возможности нет, используйте команду ср с опцией -i (сокращение от interactive). При использовании этой опции команда вначале проверяет, существует ли файл с указанным именем. Если да, то программа спросит, хотите ли вы перезаписать этот файл. Если вы ответите у, то существующий файл будет уничтожен и программа создаст новый файл в качестве копии. Если вы дадите другой ответ, он будет считаться отрицательным и выполнение команды ср будет прервано, а файл оригинала сохранен.

**ср** -i newprop proposal

Overwrite proposal? n

Для того чтобы скопировать файл из рабочего каталога в другой, нужно указать имя этого каталога команде ср в качестве второго параметра. Новая копия будет иметь то же имя, что и оригинал, но разместится в другом каталоге. Файлы в разных каталогах могут иметь одинаковые имена. Поскольку файлы находятся в разных каталогах, они считаются разными.

**ср** имена\_файлов имя\_каталога

Команда ср может использовать в качестве параметров имена многих файлов, за­данные в виде списка, поэтому можно одновременно копировать в каталог сразу не­сколько файлов. Введите имена этих файлов в командной строке, причем имя каталога должно быть последним параметром. Все эти файлы копируются в указанный каталог. В следующем примере пользователь копирует файлы preface и docl в каталог props. Обратите внимание на то, что props — последний параметр.

**ср** preface docl props

При создании списка имен файлов для команды ср или команды mv можно исполь­зовать любые групповые символы. Пусть, например, вам нужно скопировать в заданный каталог все файлы с исходными текстами программ, написанными на языке С. Вместо того чтобы указывать в командной строке все эти файлы, можно ввести групповой символ \* с расширением .с, обозначая тем самым все файлы с расширением .с (то есть все файлы исходных текстов программ С), формируя таким образом их список. В следующем примере пользователь копирует все файлы исходных текстов программ из текущего каталога в каталог sourcebks**.**

**cp** \*.c sourcebks

В следующем примере имя файла proposalменяется на versionl**.**

**mv** proposal version1

Как и при использовании команды ср, здесь очень легко ошибиться, удалив нужный файл. Переименовывая файл, вы можете выбрать имя, которое уже носит другой файл, и этот файл будет удален. Команда mv тоже имеет опцию -i, которая сначала проверяет, существует ли файл с указанным именем. Если да, программа спросит, хотите ли вы перезаписать его. В следующем примере файл с именем versionlуже существует. Про­грамма обнаруживает, что должна быть выполнена перезапись, и спрашивает, хотите вы это сделать или нет.

**ls**

Proposal versionl

**mv** -i versionl proposal

Overwrite proposal? n

Файл можно перенести из одного каталога в другой. Для этого нужно в качестве второго параметра в команде mv поставить имя каталога. В данном случае можно счи­тать, что команда mv не переименовывает файл, а просто перемещает его из одного каталога в другой. После перемещения файла у него останется то имя, которое он носил в исходном каталоге (если вы не укажете иного).

**mv** *имя\_файла имя\_каталога*

При создании списка имен файлов для команды mv можно использовать любые специальные символы. В приведенном ниже примере пользователь перемещает все файлы исходных текстов программ из текущего каталога в каталог newproj.

**mv** \*.с newproj

**Перемещение и копирование каталогов**

Подкаталоги можно копировать из одного каталога в другой. Для копирования каталога команду ср необходимо использовать с опцией -r (сокращение от recursive, то есть «рекурсивный»). Эта опция дает команде ср указание копировать каталог вместе со всеми его подкаталогами. Другими словами, копируется все поддерево каталогов, начиная с указанного. В следующем примере каталог thankyou копируется в каталог oldletters. После завершения этой операции начинают равноправно сосуществовать два подкаталога thankyou: один в каталоге letters, другой в oldletters.

**ср -r** letters/thankyou oldletters

**ls -F** letters

/thankyou

**ls -F** oldletters

/thankyou

**Удаление файла: команда rm**

В следующем примере пользователь удаляет файл oldprop.

**rm** oldprop

Команда rm может быть использована с любым числом параметров, что позволяет одновременно удалять несколько файлов. Имена этих файлов указываются в командной строке после имени команды.

**rm** proposal version1 version

Командой rm следует пользоваться осторожно, так как отменить ее действие нельзя. Если файл удален, восстановить его не удастся. Предположим, что вы случайно ввели эту команду вместо какой-то другой, например ср или mv. Когда вы опомнитесь, будет слишком поздно - файлы пропали. Чтобы избежать подобных ошибок, используйте команду rm с опцией -i, которая инициирует выдачу запроса на подтверждение уда­ления. Теперь перед удалением каждого файла система будет спрашивать, действительно та вы хотите удалить его. Если вы введете у, файл будет удален. При любом ином ответе файл не удаляется. В следующем примере посредством команды rm система получает указание удалить файлы proposal и oldprop, а затем запрашивает подтверждение по каждому из них. Пользователь решает удалить oldprop, a proposal оставить.

**rm -i** proposal oldprop

Remove proposal? n

Remove oldprop? у

Для удаления каталога и всех его подкаталогов применяется команда rm с опцией -r. Это очень мощная команда, и при ее неправильном использовании можно легко стереть все свои файлы. В следующем примере показано, как удалить каталог reportsи все его подкаталоги:

**rm -r** reports

Следующая команда стирает все файлы в текущем каталоге

**rm** \*

Это очень мощная операция, позволяющая стирать целые сегменты файловых сис­тем. Ею следует пользоваться с осторожностью.

**Присоединение дополнительной файловой системы**

Присоединение файловой системы, расположенной на запоминающем устройстве, к основному дереву каталогов называется *монтированием устройства.* Операция монтирования предполагает присоединение де­рева каталогов, находящегося на устройстве памяти, к указанному пользователем ка­талогу. Лишь после этого можно перейти в присоединенный каталог и обращаться к его файлам. Каталог, к которому вы хотите присоединить файлы, находящиеся на за­поминающем устройстве, называется *точкой монтирования.*

Монтирование файловой системы может осуществляться ***только привилегированным пользователем***, поскольку эта задача относится к функциям системного администратора.

Для того чтобы определить, сколько свободного пространства осталось в файловой системе, можно воспользоваться командой df.

Она выдает список всех файловых систем по именам устройств, сообщает их размер и точки монтирования.

**df**

Filesystem 1024-blocks Used Available Capacity Mounted on

/dev/hda3 297635 169499 112764 60% /

/dev/hdal 205380 182320 23060 89% /mnt/dos

/dev/hdc 637986 637986 0 100% /mnt/cdrom

Ниже приведена копия файла /etc/fstab**.** Первая его строка — комментарий. Все строки комментариев начинаются со знака #. Файловая система /ргос— это специальная файловая система, которую операционная система Linux использует для управления системными процессами. Никакому реальному устройству она не соответствует. Для того чтобы сделать запись в файле /etc/fstab**,** вы можете просто отредактировать его.

#<<device>> <<mountpoint>> <<filesystemtype>> <<options>> <<dump>> <<fack>>

/dev /hda1 / ext2 defaults 0 1

/dev/hdc /mnt/cdrom iso9660 ro,noauto 0 0

/dev/fd0 /mnt/floppy ext2 defaults, noauto 0 0

/proc /proc proc defaults

/dev/hda2 none swap sw

**Команды Mount и Umount**

Любую файловую систему можно также монтировать и демонтировать при помоши команд mount и umount (обе команды выполняются с правами root !).

Команда mount принимает два аргумента: имя устройства, через которое Linux по­лучает доступ к файловой системе, и каталог в файловой структуре, к которому при­соединяется новая файловая система. *Точка\_монтирования —* это каталог, к которому вы хотите присоединить файлы, находящиеся на запоминающем устройстве. *Устрой­ство* — это специальный файл, с помощью которого система получает доступ к аппа­ратным средствам. Команда mount имеет следующий синтаксис:

**mount** устройство точка\_монтирования

Файлы устройств находятся в каталогах /dev и обычно имеют сокращенные имена, которые заканчиваются номером устройства.

Чтобы к файловой системе был возможен доступ, она должна быть смонтирована. При инсталляции Linux и создании на жестком диске раздела Linux система автоматически конфигурируется на монтирование основных файловых систем при каждом запуске. Для usb и компакт-дисков такая возможность не предусмотрена, и их придется монтировать явно.

Перед тем как останавливать систему, необходимо демонтировать все смонтированные файловые системы. Основные файловые системы демонтируются автоматически. Если, однако, вы хотите заменить смонтированную файловую систему другой, сначала нужно демонтировать первую явно. Скажем, вы смонтировали файлы, содержавшиеся на usb-носителе, а теперь хотите заменить его на другой. Для этого нужно сначала демонтировать файловую систему установленной "флешки". Файловая система демонтируется командой umount. В качестве аргументов эта команда использует имя устройства и каталог, в котором оно было смонтировано. Вот синтаксис команды umount:

**umount** устройство точка\_монтирования

Демонтировать файловую систему можно и воспользовавшись именем этого каталога:

**umount** /mydir

Для команды umount установлено одно существенное ограничение. Нельзя демон­тировать файловую систему, в которой вы работаете в настоящий момент. Если вы перейдете в какой-либо каталог файловой системы и затем попытаетесь демонтировать ее, то получите сообщение об ошибке — вас уведомят о том, что файловая система занята.

**Монтирование USB-накопителей**

В прин­ципе, монтировать устройство можно в любой каталог, но лучше создать каталог для флеш-накопителей, /mnt/usb и разрешить к нему обращаться всем пользователям. В следующем примере файлы монтируются в этот каталог.

“Флешка”, как правило, может иметь файловую систему FAT32 или NTFS. Поэтому нам нужно сообщить об этом системе. Для FAT32 выполним следующую команду:

**mount** -t vfat -o rw /dev/sdc1 /mnt/usb

-t vfat – указывает о файловой системе FAT 32 носителя

/dev/sdc1 – наше монтируемое устройство (определить его имя можно воспользовавшись командой **df** найдя там обозначение монтируемого носителя типа sdX1**)**

/mnt/usb – точка монтирования

-o rw – разрешаем чтение и запись

Для NTFS соответственно:

**mount**-t ntfs -o rw /dev/sdc1 /mnt/usb

Для демонтирования используется команда umount:

**umount** /dev/sdc1

В операции umount можно указать каталог, в котором смонтирован съёмный диск.

**umount** /mnt/usb

Теперь вы можете извлечь флешку, вставить новую и затем смонтировать ее.

При останове системы производится автоматическое демонтирование всех смонтированных дисков. Явно демонтировать их не нужно.

**Монтирование компакт-дисков**

В Linux обычно каталог /mnt/cdrom зарезервирован для файловых систем компакт-писков. Строка для этого каталога имеется в файле /etc/fstab. Чтобы смонтировать компакт-диск, достаточно ввести команду **mount** и имя каталога /mnt/cdrom. Имя уст­ройства указывать не надо. После монтирования можно обращаться к компакт-диску через каталог /mnt/cdrom.

**mount** /mnt/cdrom

Чтобы поменять диск, сначала нужно демонтировать компакт-диск, установленный в дисководе, с помощью команды umount. Вот как выпол­няется демонтирование диска, который вы хотите вынуть из дисковода:

**umount** /mnt/cdrom

Если вы желаете смонтировать компакт-диск в другом каталоге, необходимо указать в команде mount имя устройства. Ниже приведен пример, в котором диск, находящийся в дисководе, монтируется в каталоге /mydir. В данном примере для компакт-диска файл устройства имеет имя /dev/hdc.

**mount** /dev/hdc /mydir

**Установление прав доступа: символы прав доступа**

Права доступа на чтение, запись и выполнение обозначаются соответственно символами r,w и x. Любое из этих разре­шений можно добавлять и удалять. Символом добавления права доступа является знак плюс, +. Символом отмены является знак минус, -. В следующем примере команда chmod добавляет право на выполнение и отменяет право на запись для файла mydata. Право на чтение не изменяется.

**chmod +x-w** mydata

Используются и другие символы, которые обозначают категории пользователей. Ка­тегории «владелец», «группа» и «прочие» обозначаются соответственно символами *и, g* и о. Категория «владелец» обозначена буквой *и* (от слова user), так как владельца можно считать пользователем. Символ категории ставится перед символами, устанавливающи­ми права на чтение, запись и выполнение. Если символа категории нет, то подразу­меваются все категории, и указанные права устанавливаются для пользователя, группы и прочих. В следующем примере первая команда chmod устанавливает право на чтение и запись для группы. Вторая команда chmod устанавливает право на чтение для прочих пользователей. Обратите внимание на то, что пробелов между спецификациями прав доступа и категорией нет. Список изменений прав доступа представляет собой одну длинную фразу без пробелов.

**chmod g+rw** mydata

**chmod o+r** mydata

Пользователь может не только устанавливать, но и отменять права доступа. В сле­дующем примере для прочих пользователей устанавливается право на чтение, а права на запись и выполнение отменяются.

**chmod o+r-wx** mydata

Есть еще один символ разрешения, *а* (от слова all), который обозначает все категории. Он действует по умолчанию. В следующем примере обе команды эквивалентны. Разре­шение на чтение задается явно символом *а,* который обозначает все типы пользователей: прочие, группа и владелец.

**chmod a+r** mydata

**chmod +r** mydata

Одна из наиболее распространенных операций с правами доступа — установление права на выполнение файла. Право на выполнение по­казывает, что файл содержит выполняемые команды и может непосредственно выпол­няться системой. В следующем примере для файла lsc устанавливается право на вы­полнение.

**chmod u+x** lsc

**Абсолютные права доступа: двоичные маски**

Вместо символов разрешений многие пользователи предпочитают применять *абсо­лютный метод.* Абсолютный метод позволяет изменять сразу все права доступа. Здесь используется двоичная маска, которая обозначает все разрешения в каждой категории. Эти три категории, по три разрешения в каждой, представлены в восьмеричном формате. В восьмеричной системе счисления все числа имеют основание 8. При преобразовании в двоичный формат каждый восьмеричный разряд превращается в три двоичных. Дво­ичное число представляет собой набор единиц и нулей. Три восьмеричных разряда числа преобразуются в три набора по три двоичных разряда в каждом. Каждый двоичный разряд определяет одну из составляющих прав доступа (напоминаем — в направлении слева направо). Если двоичный разряд — 0, разрешение отсутствует. Если двоичный разряд — l, разрешение установлено. Первый слева двоичный разряд включает и вы­ключает право на чтение, второй — право на запись, третий — право на выполнение. Например, восьмеричная цифра 6 преобразуется в двоичные цифры 110. Это соответ­ствует установлению права на чтение и запись и отменяет право на выполнение.

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

При использовании двоичной маски нужно указывать сразу три цифры, которые обеспечивают установление прав доступа одновременно' для всех трех категорий. Это делает двоичную маску менее гибкой, чем символы разрешений. Чтобы установить право на выполнение и отменить право на запись для владельца файла **mydata** с со­хранением права на чтение, необходимо использовать восьмеричную цифру 5 (101 в двоичном формате). В то же время нужно указать соответствующие цифры для группы и прочих пользователей. Если право на чтение для этих категорий сохраняется, то для каждой из них следует указать восьмеричную цифру 4 (100). Это дает вам три вось­меричные цифры, 544, которые преобразуются в двоичные цифры 101 100 и 100.

**chmod 544** mydata

**Права доступа к каталогам**

В следующем примере для каталога **thankyou** устанавливается право на чтение и выполнение по категории «группа», а право на запись отменяется. Члены группы могут входить в каталог **thankyou** и получать список его файлов, но создавать в нем новые файлы они не могут.

**chmod g+rx-w** letters/thankyou

Как и в случае с файлами, при установлении прав доступа к каталогу можно пользо­ваться восьмеричными цифрами. Для указания тех же прав, что и в предыдущем при­мере, нужно использовать восьмеричные цифры 750 (двоичный эквивалент — 111 101 000).

**chmod 750** letters/thankyou

Команда ls с опцией -l выдает список всех файлов, содержащихся в каталоге. Для того чтобы получить информацию только о самом каталоге, нужно использовать эту команду с опцией d. В следующем примере команда ls -ld выдает информацию о каталоге **thankyou.** Обратите внимание на то, что первым символом в поле прав доступа является d, а это указывает, что он представляет собой каталог.

**ls -ld** thankyou

drwxr-x 2 chris 512 Feb 10 04:30 thankyou

**Задание для закрепления практических навыков:**

1. Продемонстрировать преподавателю работу команд:

ls, cat, more, mkdir, rmdir, cd, pwd, cp, mv, rm

Объяснить преподавателю назначение их ключей и параметров.

2, Смонтировать USB-накопитель или CD и продемонстрировать работу с ним.

3. Изменить права доступа к файлам и каталогам, созданным вами при выполнении пункта 1, используя символьный и абсолютный методы указания прав доступа.