# Лабораторная работа 7. Работа с файлами

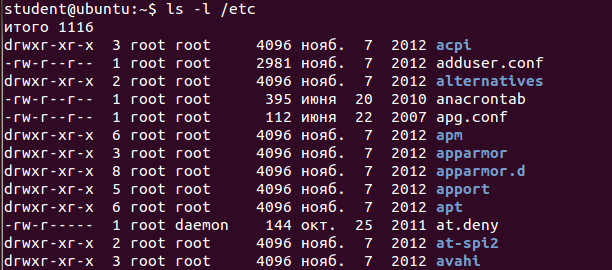
**Аннотация:** Цель работы: изучить типы и атрибуты файлов, команду поиска в Linux на примере операционной системы Ubuntu.

**Задание 7-1.** Найдите в вашей операционной системе хотя бы по одному файлу каждого типа.

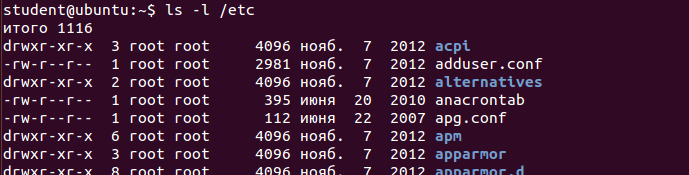
Ход выполнения:

1. Рассмотрите пример определения типа файла в Ubuntu.

Для определения типа файла используется команда *ls –l*. Например, хотим определить типы файлов, содержащихся в каталоге */etc*. Для этого вводим в командной строке терминала команду *ls –l /etc*. Фрагмент результата выполнения данной команды можно увидеть на рисунке 7.1.



*Рис. 7.1.* Определение типа файла



Каждая строка полученного результата содержит информацию об одном файле. Нас сейчас интересует первое и последнее поле каждой строки. Первый символ первого поля указывает на тип файла, а последнее поле – непосредственно имя файла.

Например, имя первого в списке файла – *abrt*, тип файла – каталог, на это указывает первый символ “*d*” первого поля *drwxr-xr-x*. Третий файл *adjtime* является обычным файлом (символ *“-”)* и т.д.

2. Ознакомьтесь с обозначениями основных типов файлов, используемых в Linux.

В Linux используются следующие типы файлов:

* Обычные файлы (-) и каталоги (**d**).



* Файлы символьного режима: эти файлы являются либо специальными системными файлами (типа /dev/null), либо периферийными устройствами, характерной особенностью которых является то, что их содержимое не буферизуется. Такие файлы обозначаются буквой **c**.



* Файлы блочного режима: эти файлы являются периферийными устройствами, и, в отличие от символьных файлов, их содержимое буферизуется. Такие файлы обозначаются буквой **b**



* Символьные или «мягкие» ссылки: они являются файлами, содержащими путь к другому файлу. Обозначаются такие файлы буквой **l**.



* Каналы: используются в командах shell'а, используются очень редко. Такие файлы обозначаются буквой **p**.
* Сокеты: это тип файла для всех сетевых подключений, но только некоторые из них имеют названия. Такие файлы обозначаются буквой **s**.



3. Найдите в вашей операционной системе файлы каждого типа.

**Задание 7-2.** Выясните экспериментальным путем отличие символьных («мягких») ссылок от «жестких» ссылок.

Ход выполнения:

1. Создайте обычный файл:

*touch myfile*

2. Запишите в него символы «*abcd*»:

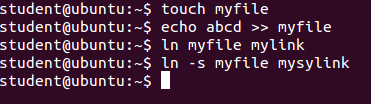
*echo abcd>>myfile*

3. Создайте «жесткую» ссылку на *mylink* на файл *myfile*:

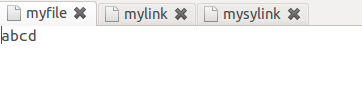
*ln myfile mylink*

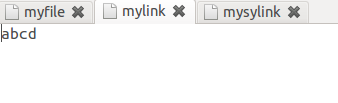
4. Создайте «мягкую» (символьную) ссылку на *mysymlink* на файл *myfile*:

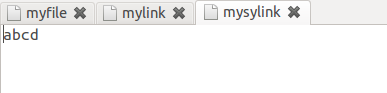
*ln -s myfile mysymlink*



5. Просмотрите и сравните содержимое всех трех созданных файлов.







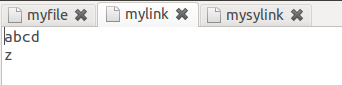
6. Допишите в файл *myfile* символ «*z*»:

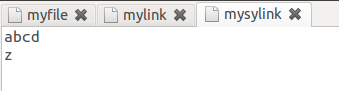
*echo z>>myfile*



Еще раз сравните содержимое всех трех созданных файлов.







7. Обратите внимание на индексные дескрипторы (inode) всех трех файлов и на количество ссылок на файлы (рис. 7.2).



**

*Рис. 7.2.* Индексные дескрипторы файлов-ссылок

Опция *-i* команды *ls* выводит номер inode в первом поле выходных данных. Третье поле представляет собой количество ссылок на inode.



8. Удалите файл *myfile*:

*rm myfile*



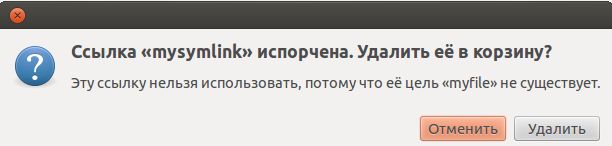
Обратите внимание на индексные дескрипторы (inode), количество ссылок у оставшихся файлов, а также на изменения, которые произошли с символьной ссылкой (рис. 7.3).



*Рис. 7.3*.Последствия удаления файла, на который были созданы ссылки

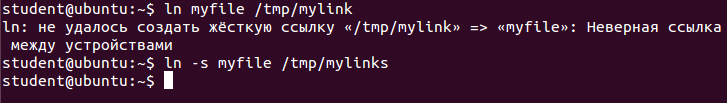


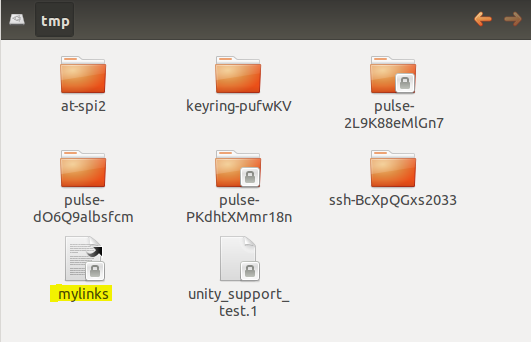
9. Просмотрите содержимое оставшихся файлов-ссылок. Сделайте выводы.



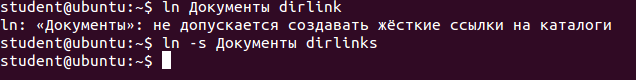
10. Попробуйте создать «мягкую» и «жесткую» ссылки на inode в каталоге, который находится в другой файловой системе. Возможно ли это? Почему?

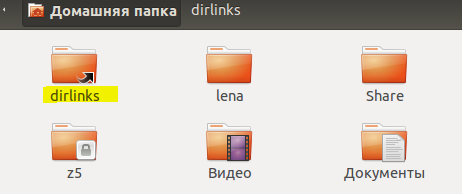






11. Попробуйте создать «мягкую» и «жесткую» ссылки на каталоги. Возможно ли это? Почему?





**Задание 7-3.** Изучите атрибуты, которые могут быть назначены файлам в Linux.

Ход выполнения:

1. Ознакомьтесь с информацией об атрибутах файлов в Linux.

Файловые системы Linux имеют свои собственные атрибуты файлов.

* **A** («no Access time»): означает, что всякий раз при обращении к файлу для чтения или записи, у него не будет обновляться время последнего доступа. Это может быть полезно, например, для файлов, к которым очень часто обращаются для чтения, особенно из-за того, что это единственный параметр в inode, который изменяется при открытии файла только для чтения.
* **a** («append only»): если для файла установлен этот атрибут, и этот файл открыт для записи, то единственной доступной операцией будет добавление данных к его предыдущему содержимому. Для каталога это означает, что вы сможете только добавить файлы, но не сможете переименовать или удалить ни одного из существующих файлов. Только root может установить или снять этот атрибут.
* **d** («no dump»): dump – это стандартная утилита для резервного копирования. Она выполняет резервное копирование любой файловой системы, для которой параметр *dump* равен 1 в файле /etc/fstab. Но если этот атрибут установлен для файла или каталога, то он, в отличие от других, будет пропущен при резервном копировании. Обратите внимание, что при установке его для каталогов, это также распространяется на все их подкаталоги и файлы.
* **i** («immutable»): файл с этим атрибутом вообще не может быть изменен: он не может быть переименован, на него не может быть создана ссылка и он не может быть удален. Только root может установить или снять этот атрибут. Обратите внимание, что это также предотвращает изменение времени последнего доступа, поэтому вам нет необходимости устанавливать атрибут **A**, если установлен **i**.
* **s** («secure deletion»): когда удаляется файл или каталог с этим атрибутом, блоки, которые он занимал на диске, перезаписываются нулями.
* **S** («Synchronous mode»): если для файла или каталога установлен этот атрибут, все его изменения синхронизируются и немедленно записываются на диск.

2. Ознакомьтесь с информацией о командах, позволяющих изменять и просматривать атрибуты файлов в Linux.

Существуют две команды для управления атрибутами файла: *lsattr* и *chattr*. Команда *lsattr* выводит список («LiSt») атрибутов, а команда *chattr* изменяет («CHange») их. Эти атрибуты могут быть установлены только для каталогов и обычных файлов.

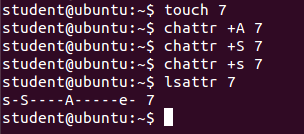
Следующая команда добавит атрибут A к файлу myfile:

*chattr +A myfile*

Чтобы удалить атрибут A у файла file, нужно написать:

*chattr –A myfile.*

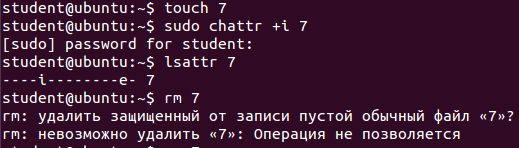
3. Создайте в своем домашнем каталоге файл. Назначьте ему атрибуты A, s и S. Что будет происходить при его изменении? При удалении?

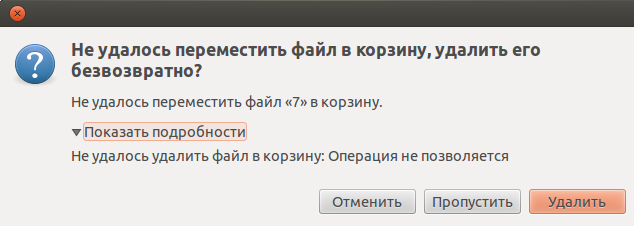


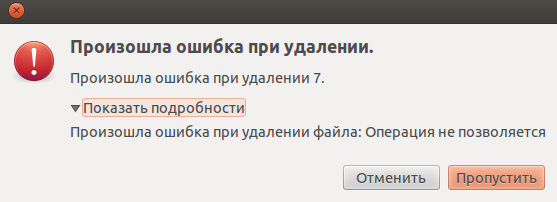
1. все его изменения синхронизируются и немедленно записываются на диск (S)
2. при удалении блоки, которые он занимал на диске, перезаписываются нулями(s)
3. что всякий раз при обращении к файлу для чтения или записи, у него не будет обновляться время последнего доступа (A)



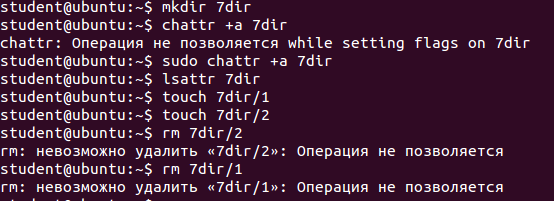
4. Какой атрибут нужно назначить файлу, чтобы защитить его от удаления? Проверьте правильность своего ответа экспериментально.

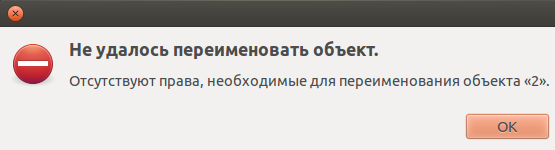






5. Создайте в своем домашнем каталоге подкаталог, в который можно будет только добавлять новые файлы. Удалять и переименовывать существующие файлы нельзя.





**Задание 7-4.** Изучите команду поиска файлов *find*.

Ход выполнения:

1. Ознакомьтесь с информацией о команде *find*.

Команда *find* может искать файлы по имени, размеру, дате создания или модификации и некоторым другим критериям.

Общий синтаксис команды *find* имеет следующий вид:

*find [список\_каталогов] критерий\_поиска*

Параметр «список\_каталогов» определяет, где искать нужный файл. Например:

*find /home/doc /usr/doc /usr/locale/doc -name instr.txt*

Началом «критерия\_поиска», определяющего, что именно должна искать программа *find*, считается первый аргумент, начинающийся на «**-**», «**(***»*, «**)**», «**,**» или «**!**». Все аргументы, предшествующие «критерию\_поиска», трактуются как имена каталогов, в которых надо производить поиск. Если не указано ни одного пути, поиск производится только в текущем каталоге и его подкаталогах.

Чаще всего поиск проводится по именам файлов, как это показано в предыдущем примере, то есть «критерий\_поиска» задается как «*-name* имя\_файла». Вместо опции *-name* можно использовать опцию *-path*, тогда команда будет искать совпадения в полном имени файла, с указанием пути. Например, команда

*find . -path './sr\*sc'*

найдет в текущем каталоге подкаталог *./src/misc*. Вместо полного имени файла или каталога в этом примере использован так называемый «шаблон имени».

2. Ознакомьтесь с информацией о шаблонах имен файлов.

Чаще всего шаблоны имен файлов строятся с помощью специальных символов «\*» и «?». Значок «\*» используется для замены произвольной строки символов. В Linux

* **«\*»** соответствует всем файлам, за исключением скрытых;
* «**.\***» соответствует всем скрытым файлам (но также текущему каталогу «**.»** и каталогу уровнем выше «**..**»);
* **«\*.\*»** соответствует только тем файлам и каталогам, которые имеют "**.**" в середине имени, или оканчиваются на точку;
* Значок «**?**» заменяет один произвольный символ, поэтому index?.htm будет соответствовать именам index0.htm, index5.htm и indexa.htm.
* В квадратных скобках [] дается либо список возможных символов, либо интервал, в который должны попадать возможные символы. Например, [abc]\* соответствует всем именам файлов, начинающимся с a, b, c; \*[I-N1-3] соответствует файлам, оканчивающимся на I, J, K, L, M, N, 1, 2, 3.

Несколько примеров простых критериев поиска перечислены в следующей таблице 7.1.

Таблица 7.1. Примеры простых критериев поиска

|  |  |
| --- | --- |
| -name шаблон | Ищет файлы, имена которых соответствуют шаблону. |
| -group имя | Ищет файлы, принадлежащие указанной группе. |
| -size число[c] | Ищет файлы, размером в число 512-байтных блоков. Если после числа стоит символ c, значит размер указан в байтах (символах). |
| -mtime число | Ищет файлы, которые в последний раз изменялись число дней назад. |
| -newer образец | Ищет файлы, которые в последний раз изменялись число дней назад. |
| -type тип\_файла | Ищет файлы указанного типа. Тип задается символом b (блок-ориентированные устройства), c (байт-ориентированные устройства), d (файл каталога), f (обычный файл), p (именованный канал) либо l (символическая ссылка). |

3. Ознакомьтесь с информацией о построении более сложных критериев поиска имен файлов.

Из простых критериев можно строить более сложные с помощью логических операций *and*, *or* или операции отрицания, знаком которой служит восклицательный знак. Например, если Вы хотите найти все файлы, имена которых оканчиваются на *.txt* и *.doc*, то критерий можно записать как

*-name \*.txt -or -name \*.doc*

Можно комбинировать любое число критериев. Если операция не указана явно, то подразумевается *-and*, то есть вместо

*-name \*.txt -and -name \*.doc*

можно записать просто

*-name \*.txt -name \*.doc*

Если применяется только одна операция *-and* или *!*, то скобки обычно можно опустить, а с операцией *-or* и в сложных выражениях скобки необходимы. Перед скобкой нужно поставить обратную косую черту, а после скобки – пробел. Например, если вы хотите найти каталог по его имени, то можно сделать это командой

*find /home/ -name doc -type d*

или (с соблюдением правил построения сложных критериев)

*find /home/ \( -name doc -and -type d \)*

В следующем примере мы ищем файлы по следующему критерию: либо имя файла оканчивается на \*.tmp, либо размер файла больше 100 Кбайт.

*find /home/kos \( \( -name \*.tmp \) -or \( -size +200 \) \)*

В последнем примере стоит обратить внимание на то, что перед значением размера стоит знак «+». Такой знак можно использовать с любым числовым параметром в критериях поиска команды find. Он означает, что нужно искать файлы, у которых значение параметра больше заданного. Соответственно, знак «-» означает, что надо искать файлы, у которых значение параметра меньше заданного. Если знак «+» или «-» отсутствует, ищутся файлы, у которых значение параметра равно заданному.

3. Ознакомьтесь с информацией о применении операций к найденным файлам.

После критерия поиска в этой команде можно сразу же задать операцию, которая будет применяться ко всем файлам, найденным по указанному критерию. Простейшим примером такой операции является команда *-print*.

*find /home/kos -name \*.tmp –print*

по которой выдаются на экран список имен всех найденных файлов с указанием полного пути к файлу. Эта операция применяется по умолчанию, то есть когда никаких операций вообще не указано (как это было во всех приведенных выше примерах).

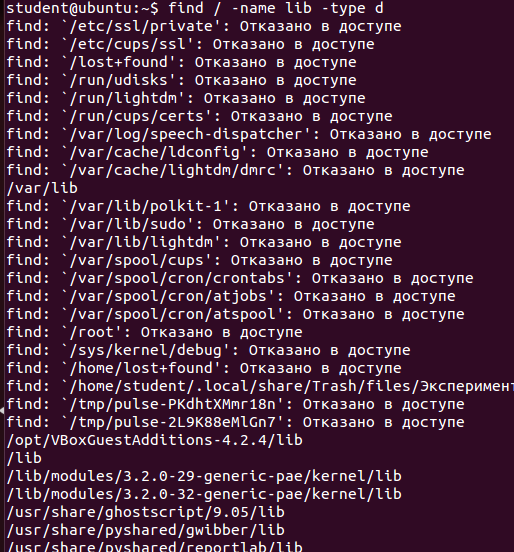
Другим примером операции, применяемой ко всем найденным файлам, может служить операция *-exec cmd {} \;*, где *cmd* - произвольная команда оболочки shell. То есть в этом случае ко всем найденным файлам (их именами заменяются поочередно фигурные скобки) применяется команда *cmd*. За *cmd* {} в этом случае должна следовать точка с запятой, экранированная обратной косой чертой. Например, если вы хотите удалить в текущем каталоге все файлы, к которым пользователи не обращались в течение 30 дней, дайте команду:

*find . -type f -atime +30 -exec rm {} \;*

Вместо *-exec* можно поставить *-ok*, тогда перед выполнением указанной команды *cmd* применительно к каждому файлу будет запрашиваться подтверждение.

Будьте очень осторожны с применением таких возможностей команды, как вызов других команд, применяемых ко всем найденным файлам. Помните, что изменения часто необратимы!

4. С помощью команды *find* найдите все папки с названием lib. Выведите их список.



5. С помощью команды *find* найдите все файлы размером не более 50 Кбайт, название которых начинается на латинскую букву “a”. Выведите их список.

