Лабораторная работа №2 «Ссылки. Права доступа»

Часть 1. Ссылки на файлы и каталоги

Виды файлов

С точки зрения пользователя в системе UNIX существует два типа объектов: файлы и процессы. Все данные хранятся в виде файлов. Работа с различными ресурсами организована через файлы. в системе существует 6 различных типов:

- обычный файл;
- каталог - это файл с именами находящихся в нем файлов и их индексных дескрипторов (inode), любой процесс может прочитать каталог при наличии прав, а записать в него может только ядро;
- специализированный файл устройства необходим для доступа к физическому устройству. Устройства делятся на символьные и блочные;
- именованный канал служит для связи между процессами;
- символическая связь (ссылка) - это еще одно обозначение путевого имени файла;
- сокет служит для обмена между процессами.

При выполнении команды ls с ключом -l выдается полная информация по файлам каталога. Первый символ в строке для любого файла определяет его тип. Команда dd предназначена для чтения данных из файла.

Жесткие и символические ссылки.

Индексный дескриптор хранит информацию о файле (атрибуты, число жестких связей, идентификаторы владельца и групп,размер файла и т.п.), кроме его имени. Между именем и inode устанавливается жесткая связь, число этих связей может быть более одной. Все жесткие связи равноправны. Изменение атрибутов или данных по одному имени автоматически распространяется на все. при удалении имени удаляется только связь, если связей больше нет удаляется весь файл.

Если необходимо создать ссылку на файл в каталоге другого пользователя, жесткая ссылка может не сработать. Это обусловлено тем, что файловую структуру ОС Linux можно физически сегментировать на файловые системы. Файловая система может располагаться на любых физических запоминающих устройствах - от дискеты до жестких дисков. Несмотря на то что файлы и каталоги во всех файловых системах присоединены к одному общему дереву каталогов, каждая файловая система физически управляет своими файлами и каталогами. Это значит, что Файл одной файловой системы нельзя связать прямой ссылкой с файлом, принадлежащим другой файловой системе.

Для решения этой проблемы применяются символические ссылки. Символическая (косвенная) ссылка содержит путевое имя файла, для которого она создается. Чтобы удалить файл, нужно удалить только прямые ссылки Если остались символические ссылки, доступ к файлу по ним будет невозможен.

При создании сиволической ссылки создается файл, имеющий тип "связь" и свой inode. В отличие от прямых ссылок, символические ссылки можно использовать для создания ссылок на каталоги. По сути дела, можно создать еще одно имя для обращения к каталогу. При этом следует помнить, что команда pwd всегда выдает фактическое имя каталога, а не символическое.

Ссылки: команда ln

С помощью команды ln файлам можно присваивать дополнительные имена. Это нужно для того, чтобы иметь возможность обращаться к файлу по разным именам из разных каталогов. Дополнительные имена часто называют ссылками.

Команда ln использует два аргумента: имя исходного файла и новое, дополнительное имя файла. В операции ls указываются оба имени, но в действительности существует лишь один физический файл.

```
$ ln исходное_имя_файла дополнительное_имя_файла
```

В следующем примере файлу today присваивается дополнительное имя weather.

```
$ ls today
$ ln today weather
$ ls today
weather
```

Файлу можно дать несколько имен, применив по отношению к нему несколько команд ln. В следующем примере файлу today присваиваются дополнительные имена weather и weekend.

```
$ ln today weather
$ ln today weekend
$ ls
today weather weekend
```

Используя команду ls с опцией -l, можно выяснить, есть ли у файла ссылки. Эта команда позволяет получить следующую информацию: права доступа, количество ссылок, размер файла и дату последнего изменения. Первое число перед именем владельца файла - это количество ссылок. Число перед датой - размер файла. В следующем примере пользователь получает полную информацию о файлах today и weather. Обратите внимание: число ссылок у обоих файлов равно двум. Более того, совпадают их размеры и даты создания. Это еще раз показывает, что указанные файлы - просто разные имена одного и того же файла.

```
$ ls -l today weather
-rw-rw-r-- 2 chris group 563 Fob 14 10:30 today
-rw-rw-r-- 2 chris group 563 Feb 14 10:30 weather
```

Данные сведения, однако, не позволяют утверждать наверняка, что имена этих файлов связаны ссылками. Вы можете просто посмотреть, совпадают ли число ссылок, размеры и даты модификации двух файлов, как в случае с файлами today и weather. Для того чтобы знать это наверняка, нужно дать команду ls с опцией -i. Эта команда сообщает имя файла и его индексный дескриптор. Индексный дескриптор - это уникальный номер, которым система обозначает конкретный файл. Если индексные дескрипторы двух имен файлов совпадают, это значит, что они относятся к одному и тому же файлу. В следующем примере пользователь получает информацию о файлах today, weather и larisa. Обратите внимание: today и weather имеют один и тот же индексный дескриптор.

```
$ ls -i today weather larlsa
1234 today      1234 weather      3976 larisa
```

Дополнительные имена, или ссылки, созданные командой ln, часто используются для обращения к одному файлу из разных каталогов. Файл, находящийся в одном каталоге, может быть связан ссылкой с другим каталогом и использоваться из него. Предположим, нужно обратиться к файлу, находящемуся в начальном каталоге, из другого каталога. Для этого нужно создать ссылку из этого каталога на файл,

находящийся в начальном каталоге. Эта ссылка будет просто еще одним именем файла. Поскольку ссылка находится в другом каталоге, она может иметь то же имя, что и оригинал. Для того чтобы создать ссылку на файл, находящийся в начальном каталоге, из другого каталога, нужно в качестве второго аргумента команды ln задать имя этого каталога.

```
$ ln имя_файла имя_каталога
```

В следующем примере в каталоге reports создается ссылка на файл today, расположенный в каталоге chris. Команда ls позволяет показать файл today в обоих этих каталогах, тогда как фактически существует лишь один экземпляр этого файла - оригинал в начальном каталоге.

```
$ ln today reports
$ ls
today reports
$ ls reports t
oday
$
```

Аналогично тому как это делалось при использовании команд cp и mv, ссылке можно дать другое имя. Для этого новое имя нужно указать через косую черту после имени каталога. В следующем примере в каталоге reports создается ссылка на файл today с именем wednesday. Фактически существует один файл, оригинал с именем today в каталоге chris, но теперь файл today связан с каталогом reports ссылкой wednesday. В этом смысле данный файл получил новое имя. В каталоге reports файл today проходит под именем wednesday.

```
$ ln today reports/wednesday
$ ls
today reports
$ ls reports
wednesday
$
```

Создавать ссылки на файлы можно с помощью путевых имен. В следующем примере для файла monday в каталоге reports создается ссылка в каталоге chris. Второй аргумент команды здесь - абсолютное путевое имя.

```
$ ln monday /home/chris
```

Для того чтобы удалить файл, нужно удалить все его ссылки. Имя файла фактически рассматривается как ссылка на этот файл. То есть с помощью команды rm удаляется именно ссылка на тот или иной файл. Если ссылок было несколько и одна из них удаляется, к файлу можно обращаться по оставшимся даже в случае удаления исходной ссылки - первоначального имени файла. В следующем примере файл today удаляется командой rm, но остается ссылка на этот файл с именем weather. К файлу можно обращаться по этому имени.

```
$ ln today weather
$ ml today
$ cat weather
The storm broke today
and the sun came out.
$
```

ОС Linux поддерживает так называемые символические ссылки. Ссылки, которые мы рассматривали до сих пор, называются прямыми ссылками (или жесткими ссылками -

hard link). В принципе, в большинстве случаев удобно использовать именно прямые ссылки, но им присущ один серьезный недостаток: если вы попытаетесь создать ссылку на файл в каталоге другого пользователя, прямая ссылка может не сработать. Это обусловлено тем, что файловую структуру ОС Linux можно физически сегментировать на файловые системы. Файловая система может располагаться на любых физических запоминающих устройствах - от дискеты до комплекта жестких дисков. Несмотря на то, что файлы и каталоги во всех файловых системах присоединены к одному общему дереву каталогов, каждая файловая система физически управляет своими файлами и каталогами. Это значит, что файл одной файловой системы нельзя связать прямой ссылкой с файлом, принадлежащим другой файловой системе. Если вы попробуете создать ссылку на файл, находящийся в каталоге другого пользователя и принадлежащий другой файловой системе, прямая ссылка не сработает.

Для решения этой проблемы применяются символические ссылки. Символическая (или косвенная) ссылка содержит путевое имя файла, для которого она создается. Это не прямая ссылка, а скорее информация о том, как найти конкретный файл. Вместо того чтобы регистрировать еще одно имя файла, как это делает прямая ссылка, символическая ссылка позволяет создать еще одно обозначение путевого имени файла.

Символические ссылки создаются командой ln с опцией -s. В следующем примере пользователь создает ссылку lunch на файл /home/george/veglist.

```
$ ln -s /home/george/veglist lunch
```

Нетрудно убедиться в различии между символической ссылкой и файлом, на который она указывает. В приведенном ниже примере пользователь получает полную информацию о lunch и /home/george/veglist с помощью команды ls -1. Первый символ в строке обозначает тип файла. Символические ссылки имеют собственный тип, обозначенный как l. Тип файла для lunch - l, т.е. это символическая ссылка, а не обычный файл. Число, стоящее после group - это размер файла. Обратите внимание: размеры разные. Размер файла lunch составляет всего 20 байтов. Это обусловлено тем, что lunch - всего лишь символическая ссылка, хранящая путевое имя реально существующего файла, занимающее всего несколько байтов. Это не прямая ссылка на файл veglist.

```
$ ls -I /home/george/vegllst lunch
lrw-rw-r--  1  chris    group  20    Feb  14  10:30  lunch->
/home/george/veglist
-rw-rw-r-- 1 george group 793 Feb 14 10:30 veglist
```

Для того чтобы удалить файл, нужно удалить только прямые ссылки. Если остались символические ссылки, доступ к файлу по ним будет невозможен. В данном случае в символической ссылке содержится путевое имя более не существующего файла.

В отличие от прямых ссылок, символические ссылки можно использовать для создания ссылок на каталоги. По сути дела, можно создать еще одно имя для обращения к каталогу. При этом следует помнить, что команда pwd всегда выдает фактическое имя каталога, а не символическое. В следующем примере пользователь создает для каталога thankyou символическую ссылку gifts. Используя ссылку gifts в команде cd, пользователь переходит в каталог thankyou.

Команда pwd выдает путевое имя каталога thankyou.

```
$ ln -s /home/chris/letters/thankyou gifts
$ cd gifts
$ pwd
/home/chris/letters/thankyou
$
```

Если вы хотите получить имя символической ссылки, его можно найти в переменной cwd. Переменная cwd - это специальная системная переменная, в которой хранится символическая ссылка каталога, если таковая существует. Содержимое переменной cwd можно получить при помощи команды echo $cwd.

```
$ pwd
/home/chris/letters/thankyou
$ echo $cwd
/home/chris/gifts
```

Часть 2. Права доступа к файлам и каталогам

Категории пользователей и действия над файлами

Для каждого файла и каталога в ОС Linux задаются права доступа, которые определяют, кто и какие операции может проводить над данным файлом. Существуют три категории пользователей, которые могут иметь доступ к файлу или каталогу: "владелец", "группа" и "прочие". Владелец - это пользователь, создавший файл. Часто пользователи объединяются в группы Например, системный администратор может объединить в группу пользователей, работающих над одним проектом. Пользователи не входящие в группу относятся к категории "прочие".

Для каждой категории пользователей существует отдельный набор прав доступа на чтение, запись и выполнение. Первый набор управляет доступом самого пользователя к его файлам. Второй набор управляет доступом группы к файлам этого пользователя. Третий набор управляет доступом прочих пользователей к файлам данного пользователя. Эти три набора прав доступа на чтение запись и выполнение для трех категорий - владельца, группы и прочих -образуют в совокупности девять типов разрешений на действия с файлом. Команда ls с опцией -l выдает подробную информацию о файле, включая права доступа к нему.

Отсутствие права доступа обозначается дефисом, -. Право на чтение обозначается буквой r, право на запись - буквой w, право на выполнение - буквой x. Первый трехсимвольный набор - это права доступа к файлу для категории "владелец". Второй трехсимвольный набор - это права доступа к файлу для категории "группа". Третий трехсимвольный набор - это права доступа к файлу для категории "прочие".

Для создания различных конфигураций прав доступа применяется команда chmod. В качестве аргументов в этой команде используются два списка: список изменений прав доступа и список имен файлов. Список изменений прав доступа можно задавать двумя способами. В первом, который называется символическим методом, используются символы r, w, x. Во втором, который называют абсолютным методом, применяется так называемая двоичная маска.

Установление прав доступа: символы прав доступа

В символическом методе 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, -. Категории пользователей "владелец", "группа" и "прочие" обозначаются соответственно символами u, g и o. 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. Есть еще один символ разрешения, a (от all), который обозначает все категории. Он действует по умолчанию.

Помимо прав на чтение, запись и выполнение можно устанавливать для пользователей право владения программами на время их выполнения. Как правило,

программа во время выполнения обладает правами того пользователя, который ее запустил, даже если сам файл программы принадлежит другому пользователю. Установка бита "смены идентификатора пользователя" позволяет остальным пользователям выполнять программу с правами ее настоящего владельца. Например, многими программами в системе владеет пользователь root, тогда как выполняют их обычные пользователи. Иногда при работе таких программ возникает необходимость изменения файлов, принадлежащих пользователю root. В этом случае обычному пользователю нужно запустить эту программу с сохранением права пользователя root, чтобы она получила право изменять принадлежащие ему файлы. Бит "смены идентификатора группы" имеет то же значение, что и бит "смены идентификатора пользователя", но только для групп. Пользователи выполняют программу с правами той группы, к которой принадлежит ее владелец. Такая программа может изменять файлы, принадлежащие группе.

Для установки бита смены идентификатора пользователя или группы используется опция s. Бит смены идентификатора пользователя или группы обозначается буквой s в позиции "выполнение" категории "владелец" или "группа". Это право фактически является вариантом права выполнения, x.

Есть еще одно специальное право доступа, которое повышает эффективность программ. Так называемый sticky-бит дает системе указание оставить программу в памяти после завершения ее выполнения. Это полезно для небольших программ, которые часто используются многими пользователями. На наличие sticky-бита указывает буква t в позиции "выполнение" категории "прочие". У программы с правом доступа на чтение и выполнение и установленным sticky-битом права доступа обозначаются как r-t.

Установление прав 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. Три восьмеричных разряда числа преобразуются в три набора по три двоичных разряда в каждом. Итого получается девять цифр, что в точности соответствует количеству разрешений доступа к файлу.

Восьмеричные цифры можно использовать как маску для установления различных прав доступа к файлу. Каждая восьмеричная цифра относится к одной из категорий пользователей. При этом категории нумеруются слева направо, начиная с категории "владелец". Первая восьмеричная цифра относится к владельцу, вторая к группе, а третья - к прочим пользователям.

Чтобы обозначить бит смены идентификатора пользователя и sticky-бит нужно ввести перед этими восьмеричными цифрами еще одну. Бит смены идентификатора пользователя обозначается цифрой 4 (100); бит смены идентификатора группы обозначается цифрой 2 (010); sticky-бит обозначается цифрой 1 (001).

Права доступа к каталогам

Права доступа можно устанавливать и для каталогов. Если для каталога установлено право на чтение, пользователи могут получать список файлов, находящихся в данном каталоге. Право на выполнение позволяет переходить в этот каталог. Право на запись дает возможность пользователю создавать и удалять файлы в этом каталоге. При создании каталога для него автоматически устанавливаются права на чтение, запись и выполнение для владельца. Это позволяет получать список файлов, содержащихся в каталоге, переходить в него и создавать в нем файлы.

Как и в случае с файлами, права доступа к каталогам задаются для владельца, группы и прочих пользователей. Также при установлении прав доступа к каталогу можно пользоваться символами прав доступа и двоичными масками.

Команда ls с опцией -ld выдает полную информацию только о каталоге.

Если у пользователя есть файлы, доступ к которым он хотел бы предоставить другим пользователям, то нужно установить права доступа не только для этих файлов, но и для каталога, в котором они находятся. Другой пользователь, желающий получить доступ к файлу, должен сначала войти в каталог, где этот файл находится. Это же касается и родительских каталогов.

Даже если к каталогу имеют право доступа другие пользователи, они смогут попасть в него лишь в том случае, если имеют право доступа и к родительскому каталогу данного каталога. Поэтому необходимо внимательно следить за деревом каталогов. Чтобы пользователь мог работать в каталоге, ему должны быть доступны все остальные каталоги, стоящие в дереве выше этого каталога.

Изменение владельца и группы: команды chown и chgrp

Доступ к файлу могут иметь все пользователи, но права доступа к нему может менять только владелец. Если же необходимо передать контроль над правами доступа к файлу другому пользователю, надо заменить владельца файла. Контроль над файлом передается другому пользователю с помощью команды chown. В качестве первого аргумента в этой команде указывается имя пользователя, которому передается контроль. После него можно дать список файлов, которые вы передаете.

С помощью команды chgrp можно изменить группу, владеющую файлом. В качестве первого аргумента эта команда принимает имя новой группы для файла или файлов. После имени группы можно дать список файлов, которые передаются в эту группу.

Задание для выполнения

Часть 1.

1. Изучить назначение и ключи команды ln.

- создать жесткую ссылку на файл. Просмотреть содержимое файла, используя ссылку. Удалить файл. Просмотреть содержимое файла. Объяснить результат;
- создать жесткую ссылку на каталог. Объяснить результат;

2. Выполнить все задания пункта 1, создавая не жесткие, а символьные ссылки.

3. Создать жесткую и символьную ссылки на файл. С помощью команды ls просмотреть inod файла и ссылок. Объяснить результат.

Часть 2.

1. Изучите при помощи man опцию -l команды ls. Просмотрите права каталогов /etc, /bin и домашнего каталога. Просмотрите права файлов, содержащиеся в этих каталогов. Выявите тенденции (файлов с какими правами в каких каталогах больше). Сделайте вывод.

2. Изучите материал, посвящённый пользователям и группам пользователей. Изучите руководство по командам chown и chgrp. Выясните, кто является владельцем и к какой группе владельцов принадлежат файлы вашего домашнего каталога, каталогов /etc, /root, /bin и /dev.

3. Определите атрибуты файлов /etc/shadow и /etc/passwd попробуйте вывести на экран содержимое этих файлов. Объясните результат.

4. Изучите команду chmod. Создайте в домашнем каталоге любые четыре файла, установите при помощи восмеричных масок на каждый из них в отдельности следующие права:

- для себя все права, для группы и остальных - никаких;
- для себя чтение и запись, для группы чтение, для остальных - все;
- для себя исполнение и запись, для группы никаких, для остальных чтение;
- для себя запись, для группы все, для остальных - только запись.

5. Выполните задание предыдущего пункта, используя в команде chmod только символы прав доступа.

6. Переведите номер своей зачетной книжки в восьмеричную систему счисления, разбейте полученное значение на группы по 2-3 цифры и создайте файлы с правами доступа, выраженными полученными масками. Сопоставьте данные маски с символами прав доступа и объясните, какие операции с данными файлами доступны каким субъектам системы.

7. В домашнем каталоге создайте файл и установите на него права так, чтобы его можно было только редактировать.

8. Скопируйте в свой домашний каталог файл ls из каталога /bin. Запретите выполнение этого файла и попробуйте выполнить именно его, а не исходный(!). Объясните результат.

9. Изучите на что влияют права доступа в случае каталогов. Попробуйте зайти в каталог /root, объясните результат и причину.

Отчет

Как и в других работах, отчет по проделанной работе представляется преподавателю в стандартной форме: на листах формата А4, с титульным листом, целью, ходом работы и выводами по выполненной работе. Каждое задание должно быть отражено в отчете следующим образом: 1) что надо было сделать, 2) как это сделали, 3) что получилось, и, в зависимости от задания – 4) почему получилось именно так, а не иначе. При наличии заданий с вариантами необходимо указать свой вариант, и расчет его номера, а также всё вышеуказанное для данного варианта задания.