top
КОМПЬЮТЕРНАЯ АКАДЕМИЯ

Основы системного администрирования и сетевых технологий

УРОК №4

1. Режим доступа к файлам и каталогам

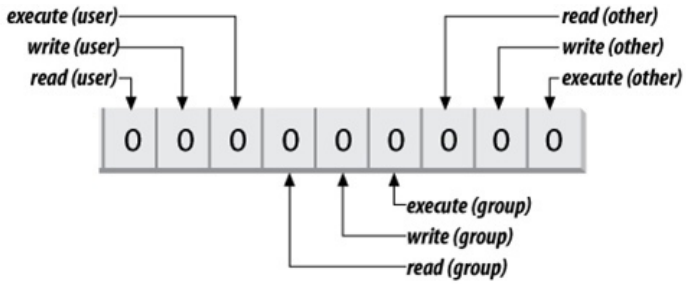

Рисунок 1. Основные биты режима доступа к файлу

- самая левая в этой тройке цифра – права владельца;
- средняя цифра – права группы;
- правая цифра – права всех остальных.

Каждая из этих восьмеричных цифр представляет собой комбинацию трех битов, каждый из которых отвечает за право на (слева направо):

- чтение (r);
- запись (w);
- исполнение (x).

В Linux доступ к файлам определяется для трех типов пользователей: владельца файла, группы владельца и остальных пользователей. Каждый из этих типов пользователей может иметь определенные права на чтение, запись и исполнение файла. Таким образом, владелец файла может иметь полный доступ к файлу, включая чтение, запись и исполнение.

Исходя из трех типов пользователей и их прав, можно определить политику доступа к файлам и каталогам. Обычно права доступа к файлу изменяются от максимальных у владельца до минимальных у остальных пользователей, вплоть до полного отсутствия прав. Установить или изменить права доступа к файлу или каталогу могут только владелец файла и суперпользователь. Для изменения прав доступа к файлу используется утилита chmod.
Права доступа к файлу (эти сведения называются режимом доступа файла - mode) или каталогу описываются тремя основными восьмеричными цифрами (рис. 1)

Если в бите установлена 1, то доступ разрешен, если 0 – запрещен. Таким образом, права доступа к файлу, описанные цифрой 644, означают, что владелец (6 - 110) может писать и читать, группа и остальные пользователи (4 - 100) – только читать.

Посмотрим, что означают чтение, запись и выполнение файлов и каталогов с точки зрения функциональных возможностей.

1. Режим доступа к файлам и каталогам

Чтение:
- просмотр содержимого файла;
- просмотр содержимого каталога.

Запись:
- изменение содержимого файла;
- удаление или перемещение файлов в каталоге.

Выполнение:
- выполнение файла (запуск программы);
- возможность поиска в каталоге в комбинации с правом чтения.

1.1 Режим доступа к файлам

Когда пользователь, программа или командная оболочка запрашивает доступ к файлу, ядро операционной системы проверяет UID объекта, который запрашивает доступ. Если UID совпадает с UID владельца файла, то ядро определяет допустимые операции на основе первых трех битов набора разрешений для данного файла. Если UID не совпадает с UID владельца файла, то ядро сравнивает GID объекта, запрашивающего доступ, с GID группы, к которой принадлежит файл. Если эти два идентификатора совпадают, то разрешения, предоставленные группе, владеющей файлом, используются для определения допустимых операций. Если ни UID, ни GID не совпадают, то запрашивающий доступ объект попадает в категорию «остальные». В этом случае разрешения, определяемые третьей группой файлов, используются для определения допустимых операций. Тема: доступ к файлам в операционной системе.

1.1 Режим доступа к файлам

Для выполнения сценария программной оболочки необходимо иметь права на чтение и выполнение соответствующего файла. Если же нужно запустить откомпилированную программу, то достаточно прав на ее выполнение. Разница заключается в том, что откомпилированные программы запускаются ядром операционной системы, и пользователь не может запустить их иным способом. В случае с сценарием оболочки, необходимо иметь возможность чтения инструкций из файла сценария. Если у пользователя нет таких прав, то он не сможет выполнить сценарий.

Однако, если у пользователя есть право на чтение файла сценария, но нет права на его выполнение, он все равно может выполнить этот файл, используя его имя в качестве параметра для интерпретатора. Тема: права доступа к файлам в операционных системах. В тексте рассматривается вопрос о необходимости прав на чтение и выполнение файлов для запуска сценариев программной оболочки и откомпилированных программ. Объясняется различие в требованиях к правам доступа для этих двух типов файлов.

Определить, чем является данный файл – исполняемым файлом, сценарием оболочки, текстовым документом или чем-то другим, можно с помощью команды file.

1.2 Режим доступа к каталогам.

Разрешения доступа для каталогов отличаются от разрешений для файлов. Каталог, как и любой другой файл, имеет владельца и группу. Но права на запись, чтение и выполнение имеют другое значение для каталогов, чем для файлов. Если у каталога есть право на чтение, то владелец, группа и другие пользователи могут просматривать содержимое каталога, то есть получать список имен файлов, находящихся в этом каталоге. Однако сами файлы внутри каталога могут быть недоступны для чтения. Тема: Разрешения доступа в операционных системах.

Разрешение на запись дает возможность пользователю записывать файлы в каталог. Чтобы создавать или изменять файлы в каталоге, нужно иметь разрешение на запись для этого каталога. Однако, это разрешение может быть недостаточным для изменения или удаления уже существующего файла. Для этого необходимы специальные разрешения, которые устанавливаются для каждого файла отдельно. Наличие права на выполнение позволяет переходить в этот каталог при помощи команды cd (делать его текущим).

Если пользователь не обладает правом выполнения каталога, это не лишает его возможности читать список файлов каталога, создавать в нем новые файлы или удалять существующие. Однако, войти в этот каталог такой пользователь не может.

1.3 Определение режима доступа

```
drwxr-xr-x 8   alex        users 4096 Dec 12 12:02 alex
drwxr-xr-x 8   caroline    users 4096 Dec 12 12:02 caroline
drwxr-xr-x 8   linda       users 4096 Dec 10 11:36 linda
drwxr-xr-x 8   sander      users 4096 Dec 10 13:22 sander
drwxr-xr-x 8   sanne       users 4096 Dec 12 11:59 sanne
drwxr-xr-x 8   stephanie   users 4096 Dec 12 12:01 stephanie
```

Рисунок 2. *Пример вывода команды ls -l*

Определить, какие права доступа к файлу установлены можно с помощью команды **ls –l** (рис. 2).

Формат вывода команды **ls –l** следующий:
- в первом столбце представлены права доступа к файлу (режим доступа файла);
- во втором столбце – число жестких ссылок на файл;
- в третьем столбце – имя владельца файла;
- в четвертом столбце – имя группы владельца файла;
- в пятом столбце – дата создания файла;
- в шестом столбце – имя файла или каталога.

Первая позиция в столбце прав доступа кодирует тип файла:
- - (тире) – файл;
- d – каталог;
- l – символическая ссылка;
- b – блочное устройство;
- c – символьное устройство;
- s – сокет;
- p – именованный канал (named pipe).

2. Модификаторы прав доступа

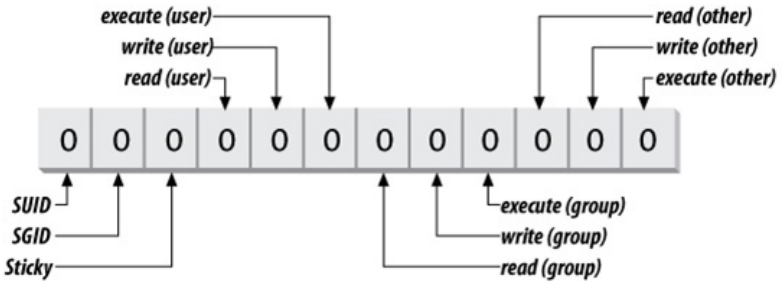

Рисунок 3. Дополнительные атрибуты режима доступа Sticky, SUID и SGID

У файлов и каталогов существуют и дополнительные атрибуты (рис. 3):
- **Sticky bit** (Save Text Attribute) – «липкий» бит;
- **SUID** (Set User ID) – установка идентификатора пользователя;
- **SGID** (Set Group ID) – установка идентификатора группы.

Эти модификаторы имеют следующее значение:
- **Sticky bit для файлов** – в современных операционных системах потерял свое значение;
- **Sticky bit для каталогов** – если установлен для каталога, то пользователь, несмотря на то, что ему разрешена запись в этот каталог, может удалять только те файлы, владельцем которых он является или к которым ему явно заданы права записи;
- **SUID для файлов** – если установлены права доступа SUID и файл исполняемый, то при запуске на выполнение, он получает не права пользователя, запустившего его, а права владельца файла. Это необходимо для того, чтобы пользователь мог работать с некоторыми системными файлами, владельцем которых является привилегированный пользователь. Например, для того, чтобы пользователь мог изменить свой пароль при помощи утилиты passwd (владелец которой - root), должен быть установлен флаг SUID, поскольку эта утилита работает с файлом /etc/passwd, изменять который может только root;

2. Модификаторы прав доступа

- **SGID для файлов** – аналогично установке бита SUID, только вместо владельца файла используется группа владельца.
- **SGID для каталогов** – файлы, которые создаются в этом каталоге, будут иметь установки группы такие же, как у каталога.

2.1 Действие атрибута Sticky bit на файлы и каталоги

Для исполняемых файлов атрибут Sticky bit (Save Text Attribute) имеет скорее историческое, чем практическое значение.
В прошлом было предписано, чтобы завершенная программа не выгружалась из памяти сразу же после ее завершения, а только после некоторого времени. Это позволяло избежать постоянной загрузки программы с диска, если она часто использовалась. Однако в настоящее время программа выгружается из памяти только в случае нехватки памяти, поэтому атрибут завершенности программы не имеет особого смысла. Атрибут завершенности программы также может быть применен к каталогу, указывая, что только владельцы имеют право изменять содержащиеся в нем файлы. Если пользователь не является владельцем файла, он не сможет его изменить.

2.2 Действие атрибутов SUID и SGID на файлы

Обычно бинарный исполняемый файл выполняется от имени вызвавшего его пользователя, то есть имеет те же самые привилегии и ограничения, что и пользователь. Однако если у исполняемого файла установлен атрибут **SUID** (Set User ID), то независимо от того, кто запускает программу, эта программа всегда будет выполняться от имени своего владельца. Как правило, таким владельцем является суперпользователь и атрибут **SUID** устанавливается для того, чтобы обычный пользователь мог запускать программы, которым для выполнения их функций необходимы привилегии суперпользователя. Этот атрибут может понадобиться и обычному пользователю, как способ сделать доступной для остальных программу, которой иначе может пользоваться только он.

В отношении **файла сценария** установка атрибута **SUID** ничего не меняет. Сценарий, как и прежде, будет выполняться от имени вызвавшего его пользователя. Чтобы добиться **SUID** эффекта от сценария, можно установить атрибут **SUID** на все вызываемые им программы.

Файл, для которого установлен атрибут **SGID** (Set Group ID) всегда выполняется от имени группы, которая обладает этим файлом. Для пользователя, который хочет сделать доступной для остальных свою личную программу, применение атрибута **SGID** часто оказывается лучшим решением, чем использование атрибута **SUID**.

2.2 Действие атрибутов SUID и SGID на файлы

Для каталогов **SGID** имеет иное значение. Если у каталога установлен атрибут, **SGID** то любому файлу, созданному в нем, в качестве группы-владельца назначается группа этого каталога, а не группа, создавшего файл пользователя.

2.3 Действие атрибутов SGID на каталоги

Узнать о том, какие дополнительные права доступа к файлам и каталогам установлены, позволяет команда **ls -l**. Набор базовых разрешений на доступ (чтение, запись, выполнение) обозначается символами «rwx». Однако если для файла установлен атрибут **SUID** или **SGID**, то вместо символа «x» в тройке будет указан символ «s». При назначении атрибута **SUID** символ «s» займет место символа «x» в тройке прав владельца файла, а для атрибута **SGID** – в тройке прав группы файла.
Атрибут **Sticky bit** отображается иначе: ему соответствует символ «t» на месте символа «x» в тройке прав для остальных пользователей.

3. Установка режима файлов и каталогов

Управлять доступом к файлам и каталогам позволяют команды **chown** (сменить владельца) и **chmod** (изменить режим). Первая из них меняет владельца файла, а вторая – режим доступа к файлу. Изменять владельца файла и режим доступа к файлу может только пользователь, обладающий соответствующими правами на доступ к файлу или каталогу.

3.1 Команда chown – изменение владельца и группы

Синтаксис:
chown [options] user-owner files
chown [options] user-owner. files
chown [options] user-owner.group-owner files
chown [options] .group-owner files
chown [options] --reference=rfile files
Часто используемые опции:
-R - рекурсивный режим, спускается вниз по иерархии каталогов до файлов и модифицирует их;
-v - выводит подробный отчет о действиях для всех файлов;
-c - аналогичен –v, но отображает только изменения.
Команда **chown** принимает в качестве аргументов имя владельца и/или группы, которые устанавливаются как владельцы, за которыми следует имя файла или каталога, для которого их необходимо установить.
Например, команда
chown sapr.stu a.txt
делает владельцем файла a.txt пользователя sapr и группу stu. Если требуется изменить только имя владельца или группы, в команде можно указать только sapr или .stu.
Команда
chown -v sapr.stu a.txt
изменит владельца и группу и выведет сообщение:
owner of a.txt changed to sapr.stu

3.2 Команда chgrp – изменение группы

Синтаксис:
chgrp [options] group-owner files
chgrp [options] --reference=rfile files
Опции команды chgrp аналогичны опциям команды chown.
Команда изменяет группу-владельца файлов на group-owner.
В случае использования первой формы синтаксиса устанавливается группа-владелец файла.
При использовании второй формы синтаксиса группа rfile используется как шаблон и применяется к файлам.

3.3 Команда chmod – изменение режима доступа к файлу

Команда chmod используется для изменения режима доступа к объектам. Эту команду можно использовать для указания прав доступа к объекту как в числовом виде (абсолютное задание режима доступа), так и в символьном виде (применяется для изменения относительно текущего состояния).
В случае с *числовым представлением режима доступа* команда выглядит следующим образом:
chmod права файл
Права передаются в виде четырех восьмеричных чисел.
1) Первое число определяет комбинацию дополнительный атрибут и может принимать значения, полученные как сумма следующих чисел:
- 1 – установка атрибута **Sticky bit**;

3.2 Команда chgrp – изменение группы

– 2 – установка атрибута **SGID**;
– 4 – установка атрибута **SUID.**
То есть, число 7 означает, что необходимо установить все дополнительные атрибуты (1+2+4), число 6 (2+4) означает, что устанавливаются атрибуты **SGID** и **SUID**, а атрибут **Sticky bit** не устанавливается и т.д. Если никакие биты дополнительных атрибутов устанавливать не нужно, то первое число можно пропустить.
2) Второе число определяет права владельца. Оно может принимать значения от 0 до 7 включительно:
– 0 – запрещено все;
– 1 – разрешено выполнение;
– 2 – разрешена запись;
– 3 – разрешено запись и выполнение;
– 4 – разрешено чтение;
– 5 – разрешено чтение и выполнение;
– 6 – разрешено чтение и запись;
– 7 - разрешено все.
3) Третье число определяет права группы. Может принимать значения от 0 до 7, аналогичные значениям для владельца.
4) Четвертое число определяет права остальных пользователей. Может принимать значения от 0 до 7, аналогичные значениям для владельца.
Например, команда
chmod 771 text
установит для файла text следующие права:
1) атрибуты **Sticky bit**, **SUID** и **SGID** – не установлены, поскольку первое из четырех чисел не указано (опущено), следовательно, оно равно 0. Опускать можно только первое

3.2 Команда chgrp – изменение группы

число и только в том случае, если его значение равно 0. Команду можно было бы выполнить и так:

chmod 0771 text

2) Число 771 определяет права
rwxrwx-—x, то есть владелец и члены его группы могут делать с файлом всё, а остальные пользователи только выполнять.

В символьном режиме команда **chmod** имеет формат
chmod параметры права файл

Параметры могут включать комбинацию следующих значений:

– u – изменить права владельца;
– g – изменить права группы;
– o – изменить права остальных пользователей;
– a – изменить все права (то же самое, что передать значение ugo).

Перед указанием прав можно задать режим их изменения относительно существующих:

– + - добавить;
– - - удалить;
– = - заменить новыми (старые значения будут уничтожены).

После этого устанавливается режим доступа:

– r – чтение;
– w – запись;
– x – выполнение;
– X – выполнение, если файл является каталогом;
– s – SUID или SGID;
– t – Sticky bit (в этом случае только владелец файла и каталога сможет удалить его);

3.2 Команда chgrp – изменение группы

– u – установить права всем пользователям, как у владельца;
– g - установить права всем пользователям, как у группы;
– o - установить права всем пользователям, как у «остальных пользователей».
Например, команда
chmod g-r text
отменит возможность чтения файла text у группы.
Чтобы назначить файлу text атрибут SUID, оставив остальные атрибуты без изменения, необходимо выполнить команду
chmod u+s text.
Команда
chmod g+w,o-r text
добавляет разрешение на запись для группы и удаляет разрешение для чтения для остальных.

4. Режим доступа по умолчанию

Каждый файл должен иметь владельца, группу и режим доступа. Поэтому эти три элемента по тем или иным правилам всегда назначаются каждому создаваемому файлу. Обычно владельцем созданного файла назначается создавший его пользователь. В качестве группы, владеющей файлом, назначается группа входа в систему этого пользователя. Режим доступа к файлу, назначаемый ему по умолчанию, определяется при помощи значения пользовательской маски (umask - user mask) пользователя.

Значение **umask** – это восьмеричное число, которое побитно вычитается из числа 0777 или 0666 (в зависимости от типа файла), в результате получается набор разрешений на доступ, назначаемый создаваемому файлу по умолчанию. Тип файла влияет на начальный режим доступа следующим образом: если файл является бинарным исполняемым, значение umask побитно вычитается из 0777, для всех остальных файлов значение umask побитно вычитается из 0666.

Например, если значение umask равно 022, то бинарным исполняемым файлам по умолчанию назначается режим доступа 755

0 0 0 1 1 1 1 1 1 1 1 1

-

0 0 0 0 0 0 0 1 0 0 1 0

---

0 0 0 1 1 1 1 0 1 1 0 1 (rwxr-xr—x), а всем остальным файлам назначается режим доступа 644 (rw-r--r—-).

4. Режим доступа по умолчанию

Определить текущее значение **umask** можно, выполнив команду umask без аргументов. Для изменения значения umask нужно вызвать эту команду с новым значением в качестве аргумента, например:
umask 222

5. Пример: создание общего каталога для группы пользователей

Пусть необходимо создать общий рабочий каталог для бригады студентов, выполняющих практическое задание. Каталог будет называться brigada3 и располагаться в /home. В бригаду входят пользователи с регистрационными именами alex, vera и ivan.
Все члены бригады имеют полный доступ к файлам, но только создатели файлов в /home/brigada3 могут их удалять. Пользователи, не являющиеся членами бригады, не имеют доступа к файлам.
1. Создание группы brigada3:
groupadd brigada3
2. Добавление ранее зарегистрированных пользователей к группе:
usermod –a -G brigada3 alex
usermod –a -G brigada3 vera
usermod –a -G brigada3 ivan
3. Создание каталога для бригады:
mkdir /home/brigada3

4. Установка группы-владельца созданного каталога:
chgrp brigada3 /home/brigada3
5. Защита каталога от пользователей, не являющихся членами бригады:
chmod 770 /home/brigada3
6. Установка SGID для того, чтобы гарантировать, что группа brigada3 будет владельцем всех новых файлов в каталоге /home/brigada3. Установка sticky bit для защиты файлов от удаления остальными пользователями:
chmod g+s,o+t **/home/brigada3**
7. Проверка:
su - alex
$ **cd /home/brigada3**
$ t**ouch a.txt**
$ **ls -l a.txt**
-rw-rw-r-- 1 alex brigada3 …… a.txt
$ **exit**
# **su** - vera
# **cd /home/sales**
# **rm a.txt**
rm: cannot unlink ‘a.txt’: Operation not permitted chgrp

top
КОМПЬЮТЕРНАЯ АКАДЕМИЯ

Основы системного администрирования и сетевых технологий