Назначение прав доступа к файлам и каталогам в ОС семейства Unix.
Каждый файл или каталог принадлежит и пользователю, и группе. Тем не менее это не означает, что все пользователи или члены группы имеют одинаковые права доступа.
Рассмотрим права доступа и владения для набора файлов. Для получения листинга необходимо запустить команду **1s** с опциями **-1** (подробный вывод) и **-a** (выводить все файлы, включая скрытые, т.е. файлы, имена которых начинаются с точки):
Листинг 10.1 Набор файлов с правами доступа и владельцами
 # ls -la/home/frank
total 3126
drwxr-xr-x 3 frank users 512 May12 2000
drwxr-xr-x 52 root users 9216 Mar 7 13:37
-rw-r--r-- 1 bob users 291090 Jan23 2000 l.bmp
-rw-rw-r-- 1 bob bob 2703 Dec22 1998 contents.html
В этом разделе вы узнаете, что означает строка **drwxr-xr-x,** задающая *режим доступа* к файлу.
Для каждой разновидности пользователей — *пользователь (**u**ser), группа (**g**roup)* и *другие (**o**thers)* — существует набор битов полномочий. Эти биты предоставляют возможность читать файл, модифицировать его и выполнять, иначе говоря предоставляют три вида доступа: *чтение (**r**ead), запись (**w**rite)* и *выполнение (e**x**ecute).* Смысл этих битов для файлов следующий:
r — файл можно читать;
w — файл можно модифицировать, удалять и переименовывать;
x — файл можно выполнять.

Итак, права доступа к файлу определяется записью типа **-rwxrw-r-- .** Первый дефис означает, что данный объект является файлом, следующие три символа (**rwx**) указывают права владельца файла, символы **rw-** определяют права членов группы, к которой принадлежит пользователь, последние три символа (**r--)** относятся к правам всех остальных пользователей. По умолчанию к файлам применяется режим доступа **rw-r--r--.**
Необходимо понимать, что пользователь может читать принадлежащие ему файлы лишь в том случае, если права доступа установлены соответствующим образом (т.е. разрешают чтение владельцу файла). То же самое относится и к группе.
Пользователь может удалять файлы, принадлежащие другому пользователю, только в одном единственном случае: ***если он владеет каталогом, в котором эти файлы находится.***

Изменение прав владения файлом с помощью команды chown

Суперпользователь имеет возможность изменить права владения любым файлом или каталогом в системе. Это одно из тех действий, выполнять которые может только **root:** обычные пользователи не могут передавать свои файлы друг другу. Для изменения хозяина файла применяется команда **chown** (change owner — изменить владельца):
 # chown bob file.txt
Она изменяет имя *владельца* файла **file.txt** (но не группу-владельца) на bob. Если вы помните, раньше им был frank, но теперь только bob может читать и записывать в файл, а frank может только читать его.
Команда **chown** применима и к каталогам:
 # chown bob /home/frank
Следует отметить, что пользователь может изменять файлы, которыми он владеет, в каких бы каталогах они ни находились.

Каждый администратор должен знать полезную опцию команды **chown: -R.** Она заставляет команду выполняться рекурсивно, т.е. для текущего каталога, всех файлов *в* нем и всех файлов во всех подкаталогах, находящихся ниже в иерархии. Эта опция применяется, например, при воссоздании учетной записи, когда необходимо изменить права владения всеми пользовательскими файлами:
 # chown -R bob /home/frank

Изменение групповых прав владения файлом с помощью команды chgrp

chgrp. Она изменяет права владения группы, а не пользователя. Действует она подобным образом:
 # chgrp users contents.html
После этого права владения файлом **contents.html** выглядят так:
-rw-rw-r-- 1 bob users 2703 Dec 22 1998 contents.html

Поскольку и пользователь, и группа имеют право записи, в данной ситуации любой пользователь из группы users может записывать в файл точно так же, как и bob. По умолчанию во FreeBSD для каждого пользователя создается новая группа: например, группа bob является первичной для пользователя bob. Все файлы создаются с правами владения: пользователь bob, группа bob. Если другой пользователь (например, frank) принадлежит к группе bob, он может записывать в файлы. Это механизм, который предоставляет пользователям bob и frank одинаковые возможности доступа к файлам. **chgrp** — это просто другой способ запуска команды **chown;** при желании можно использовать следующий синтаксис:
 # chown bob.users contents.html
После этого файлом **contents.html** владеют пользователь bob и группа u users. Чтобы изменить только группу владельца применяется формат:

 # chown .users contents.html
И chown, и chgrp поддерживают опцию **-R**, как было рассказано ранее.

Права доступа к файлам и каталогам
Прежде всего, каталоги распознаются по первому биту в строке — d. Это просто флажок, не связанный с правами доступа. Полномочия на работу в каталоге действуют по тому же принципу, что и полномочия на файлы, но здесь есть отличия:
r — каталог можно читать (выполнить команду **ls**);
w — каталог можно модифицировать (создавать или удалять файлы), удалять и переименовывать;
x — в каталоге можно выполнять операции над файлами, в том числе производить поиск файлов в нем.
Так, например, запись **drwxr-xr-x** означает, что данный объект является каталогом (**d**), его владелец может выполнять в этом каталоге любые действия, а его группа и остальные могут только читать и выполнять поиск. **Обратите внимание:** чтобы содержимое каталога можно было просмотреть, у него должен быть установлен бит выполнения — **x**. Для каталогов этот бит имеет смысл "поиск". Если выполняемый файл является сценарием, то пользователь для выполнения этого файла должен иметь право на чтение и выполнение. Для выполнения двоичного файла достаточно иметь только разрешение на выполнение.

Изменение прав доступа к файлам и каталогам с помощью команды chmod
Для изменения прав доступа к файлам или каталогам используется команда **chmod** (change mode — изменить режим). Команду **chmod** можно использовать двумя способами: с числовыми или символическими аргументами.

Изменение режимов с численными аргументами
Самый простой способ изменения прав доступа заключается в установке трехзначного восьмеричного числа, которое уникальным образом задает права доступа для каждого типа владельца.
Бит Значение
0 нет доступа
1 доступ на выполнение (для каталогов — поиск)
2 доступ на запись
4 доступ на чтение
Таким образом, режиму "чтение и запись" соответствует **6**, режиму "чтение и выполнение" — **5**, а режиму "чтение, запись и выполнение" **-- 7**. Комбинация цифр формирует трехзначное число, задающее стандартные права доступа к файлу. Несколько примеров приведено в табл. 10.3.

Таблица 10.3 Права доступа в численной форме *Режим Значение*
755 чтение/запись/выполнение для владельца, чтение/выполнение для группы и
остальных
644 чтение/запись для владельца, только чтение для группы и остальных
600 чтение/запись для владельца, для группы и остальных доступа нет

Применить права доступа к файлу или каталогу позволяет команда вида:
 # chmod 755 testscript.sh

Есть еще четвертая цифра, которая управляет "дополнительными" свойствами: задает особое поведение файлов и каталогов при определенных обстоятельствах. Ниже приведены значения битов, формирующих четвертую цифру.
0 — обычные права доступа.
1 — бит устойчивости. Он устанавливается только для каталогов:
владелец имеет право удалять или переименовывать только файлы, которыми он владеет, причем лишь при наличии права на запись в этот каталог.
2 — установить идентификатор группы, **setgid.** Когда такой бит установлен для выполняемого файла, последний выполняется с правами группы, владеющей файлом, а не с правами пользователя, запустившего его.
4 — установить идентификатор пользователя, **setuid.** Когда такой бит установлен для выполняемого файла, последний выполняется с правами пользователя, владеющего файлом, а не с правами того, кто его запустил. Четвертая цифра принадлежит наибольшему разряду (другими словами, она находится слева). В предыдущем примере права доступа **755** эквивалентны **0755**. Значение дополнительного бита формируется из тех же соображений, что изложены выше, поэтому, например, **3755** создает каталог с установленным битом устойчивости и битом **setgid** в дополнение к обычным правам доступа **755**.

Изменение режимов с символическими аргументами
Хотя восьмеричная система является достаточно прозрачной, существует и другой способ, легче запоминающийся. Это символический метод. Вместо указания числа в ар-гументе команды **chmod** ей задается от одного до трех флажков. Строку, содержащую их, можно отформатировать различными способами. Здесь указаны наиболее распространен-ные из них.
ПРИМЕЧАНИЕ
Полное описание гибкого синтаксиса команды **chmod** содержится в справочном руковод-стве **man chmod.**

Примеры символических режимов приведены в табл. 10.4. Каждый из них задается строкой символов. Первые символы задают, чьи права меняются (**u** — владелец, **g** -группа, **o** —остальные, **a** — всех (устанавливается по умолчанию, если первый символ не задан)). Второй символ — вид изменения (+,-,=), а третий — биты прав доступа.

Таблица 10.4 Символические режимы прав доступа

Строка режима Значение

go+w добавить право на запись для группы и остальных пользователей

+x добавить право на выполнение для всех

o-r удалить право на чтение для остальных пользователей

ugo=rw установить всем права на чтение и запись

a=rw совпадает с ugo=rw

+t добавить бит устойчивости (sticky bit)

+s добавить биты **setuid и setgid**

 # chmod g+w file.txt

Символический метод проще запоминается и поэтому удобнее для большинства операций с командой **chmod.**

СОВЕТ

Как и команды **chown и chgrp, chmod** поддерживает опцию **-R**.