Владельцы файлов

Файлы в UNIX имеют два типа владельцев:
пользователя (user owner) и
группу (group owner).

Группой называется определенный список пользователей системы.

Пользователь системы может быть членом нескольких групп :
одна из которых является *первичной (primary)*,
остальные - *дополнительными (supplementary).*

При этом что владелец-пользователь может не являться
членом группы, владеющей файлом.

Это дает большую гибкость в организации доступа к файлам.
Совместное пользование файлами можно организовать
практически для любого состава пользователей,
создав соответствующую группу и установив для нее права
на соответствующие файлы.
То есть для того чтобы некий пользователь получил доступ к этим файлам,
достаточно включить его в группу-владельца.

На этапе создания файла его владельцем-пользователем становится тот
пользователь, который его создает.

Определить, владельцев файлов можно, исполнив команду ***ls -l*** .
Вид содержимого текущего каталога:

1	2	3	4	5	6	7	8
- rw- r- - r--	*1*	*andy*	*group*	*235520*	*Dec 22*	*19:15*	*file1.txt*
- rw- rwx r--	*1*	*andy*	*student*	*3450*	*Nov 12*	*19:15*	*file2.txt*

Третья и четвертая колонки содержат имена владельца-пользователя и
владельца-группы.

Владение файлом определяет тот набор операций,
который пользователь может совершить с файлом.

Изменение владельца файла или прав доступа к файлу
может осуществлять только владелец файла или суперпользователь.

Права доступа к файлу контролируют такие операции,
как чтение, запись и запуск на выполнение
(для исполняемых файлов).

Права доступа к файлам

В Linux существуют три базовых *класса доступа* к файлу,
в каждом из которых установлены соответствующие права доступа:

User access (***u***) для владельца-пользователя файла

Group access (***g***) для членов группы, являющейся владельцем файла

Other access (***o***) для остальных пользователей .

UNIX поддерживает *три типа прав доступа* для каждого класса:
на чтение (read), обозначается символом ***r*** ,
на запись (write), обозначается символом ***w*** ,
на выполнение (execute), обозначается символом ***x*** .

Отсутствие прав доступа отображается символом **-** .

Например, права доступа к файлу a.out :

- rwx r-x r-- 1 andy student 4889 Nov 10 19:15 a.out

Тип файла	***Права владельца-пользователя***	***Права владельца-группы***	***Права остальных пользователей***
Обычный	*Чтение, запись, выполнение*	*Чтение и выполнение*	*Только чтение*

Права доступа к файлу могут быть изменены
только владельцем файла или
суперпользователем (superuser) .

Для этого используется команда ***chmod*** .

$ chmod u+w, og+r- w file1.txt file2.txt

Добавить право записи для владельца,
и право на чтение для группы и остальных.
Отключить право на запись для всех пользователей,
исключая владельца.

Права доступа для *каталогов* не столь очевидны.
Это связано с тем, что система трактует операции *чтения* и *записи*
для каталогов отлично от остальных типов файлов.

Право чтения каталога позволяет получить имена (и *только* имена)
файлов, находящихся в данном каталоге.

Чтобы получить дополнительную информацию о файлах каталога
(например, подробный листинг команды ***ls -l***)
системе придется "заглянуть" в метаданные файлов,
а это уже требует для каталога права на выполнение.

Право на выполнения также потребуется для каталога, в который
требуется перейти (т. е. сделать его текущим, с помощью команды ***cd***).

Это же право нужно иметь для доступа ко всем каталогам на пути
к целевому.
Например, если установить право на выполнения для всех
пользователей в одном из своих подкаталогов, он все равно останется
недоступным, пока ваш домашний каталог не будет иметь такого же права.

Права ***r*** и ***x*** действуют независимо,
право ***x*** для каталога не требует наличия права ***r*** и наоборот.
Комбинацией этих двух прав можно добиться интересных эффектов,
характерных именно для Linux,
например, создания *"темных"* каталогов,
файлы которых доступны только в случае,
если пользователь заранее знает их имена,
поскольку получение списка файлов таких каталогов запрещено.

$ ***pwd***	Где мы находимся?
/home/andrei	
$ ***mkdir*** *darkroom*	Создадим каталог
$ ***ls -l***	Получим его атрибуты
d rwx r- - r- - 2 group 65 Dec 22	*19:15 darkroom*
$ ***chmod*** *a-r+x darkroom*	Превратим его в *"темный"* каталог
$ ***ls -l***	Получим его атрибуты
d -wx --x --x 2 group 65 Dec 22	*19:15 darkroom*
$ ***cp*** *file_1 darkroom*	Поместим в каталог darkroom некоторый файл
$ ***cd*** *darkroom*	Перейдем в этот каталог
$ ***ls -l*** *darkroom*	Попытаемся получить листинг
##permission denied	каталога. Увы...
$ cat file_1	Тем не менее, заранее зная имя
Ok	файла можно работать с ним

(например, прочитать, если есть соответствующее право доступа)

Особого внимания требует право на *запись* для *каталога*.
Создание и удаление файлов в каталоге требуют изменения его
содержимого, и, следовательно, права на запись в этот каталог.
Самое важное, что при этом не учитываются права доступа
для самого файла.
То есть для того, чтобы удалить некоторый файл из каталога,
не обязательно иметь какие-либо права доступа к этому файлу,
важно лишь иметь право на запись для каталога,
в котором находится этот файл.

Именование файлов

Использование расширения имени файла необязательно.
Существуют соглашения по присвоению расширений.

Расширения текстовых файлов и файлов с исходниками программ — стандартны, как и в любой другой ОС:
*.txt , *.odt , *.pdf , *.c , *.cpp и т.д.

Объектные библиотеки - *.a , *.so
Исполняемые файлы - *.out , *.o
Скрытые файлы - имена начинаются с точки - .*

Первые два элемента в любом каталоге адресуют
сам этот каталог (текущий каталог) под именем " . "
и родительский каталог под именем " .. "