

Права доступа.

Вид занятия: лекция, практическое занятие.

Учебные вопросы:

1. Распределение прав доступа в Linux. Чтение. Запись. Выполнение.
2. Особенности прав у каталогов.
3. Назначение прав доступа. Команды `chmod`, `chown`, `chgrp`.
4. Sticky bit.

Время: 90 минут

Ход занятия.

1. Для каждого объекта в файловой системе Linux существует набор прав доступа, определяющий взаимодействие пользователя с этим объектом. Такими объектами могут быть файлы, каталоги, а также специальные файлы (например, устройства) — то есть по сути любой объект файловой системы. Так как у каждого объекта в Linux имеется владелец, то права доступа применяются относительно владельца файла. Они состоят из набора 3 групп по три атрибута:

- чтение(r), запись(w), выполнение(x) для **владельца**;
- чтение, запись, выполнение для **группы** владельца;
- чтение, запись, выполнение для **всех остальных**.

Такие права можно представить краткой записью:

`rwXrwxrwx` – разрешено чтение, запись и выполнение для всех

`rwXr-xr-x` – запись разрешена только для владельца файла, а чтении и выполнении для всех.

`rw-rw-r--` – запись разрешена для владельца файла и группы владельца файла, а чтение – для всех.

Такое распределение прав позволяет гибко управлять ресурсами, доступными пользователям.

2. Права доступа распространяются и на каталоги. Они означают:

`r` – если установлено право на чтение из каталога, то можно увидеть его содержимое командой `ls`.

`w` – если установлено право записи в каталог, то пользователь может создавать и удалять файлы из текущего каталога. Причем удалить файл из каталога пользователь может даже если у него **нет прав на запись** в файл. Есть возможность исправить эту ситуацию. Об этом позже.

`x` – если установлено право исполнения на каталог, то пользователь имеет право перейти в такой каталог командами наподобие `cd`.

Таким образом появляется возможность создания так называемых “скрытых” каталогов, когда невозможно получить список файлов, но пользователь точно знающий имя файла может скопировать его из “скрытого” каталога.

3. Для распределения прав доступа в Linux существует множество команд. Основные из них – это `chmod`, `chown` и `chgrp`.

Команда **chmod** (*Change MODE* – сменить режим) – изменяет права доступа к файлу. Для использования этой команды также необходимо иметь права владельца файла или права root. Синтаксис команды таков:

```
chmod mode filename, где
filename – имя файла, у которого изменяются права доступа;
mode – права доступа в символьном или абсолютном виде.
```

В символьном виде использование команды **chmod** будет выглядеть следующим образом:

```
          |r|
          |u|   |w|
          |g|  |+|  |x|
chmod |o|  |-|  |X| filename,
          |a|  |=|  |u|
          |g|
          |o|
```

где:

u, g, o, a – установка прав для пользователя, группы, остальных пользователей, всех групп прав доступа соответственно.

+, -, = – добавить, удалить, установить разрешение соответственно.

r, w, x, X, u, g, o – право чтения, записи, выполнения, выполнения если есть такое право еще у какой либо из групп доступа, такие же как у владельца, такие же как у группы, такие же как у остальных пользователей.

filename - Имя файла, у которого изменяются права.

Просмотр разрешений, установленных на файл осуществляется командой **ls** с ключом **-l**:

```
[user]$ ls -l text.txt
-rw----- 1 student student 39 Nov 19 6:57 text.txt
[user]$ chmod g+rw text.txt
[user]$ ls -l text.txt
-rw-rw---- 1 student student 39 Nov 19 6:58 text.txt
[user]$ chmod o=u text.txt
[user]$ ls -l text.txt
-rw-rw-rw- 1 student student 39 Nov 19 6:59 text.txt
[user]$ chmod o-w text.txt
[user]$ ls -l text.txt
-rw-rw-r-- 1 student student 39 Nov 19 7:00 text.txt
```

Для использования абсолютного режима необходимо представить права доступа к файлу в виде 3-х двоичных групп. Так например:

gwx r-x r-- будет выглядеть как:

```
111 101 100
```

Теперь каждую двоичную группу перевести в 8-ричное число:

```
111 ~ 7, 101 ~ 5, 100 ~ 4.
```

Чтобы задать файлу такие права необходимо выполнить команду:

```
[user]$ ls -l text.txt
-rw-rw-r-- 1 student student 39 Nov 19 7:00 text.txt
[user]$ chmod 755 text.txt
[user]$ ls -l text.txt
-rwxr-xr-- 1 student student 39 Nov 19 7:00 text.txt
```

Задание для обучающихся: попробовать изменить права файлу text.txt и задать следующие: r w x r - - r - - (744), r - - - w - - - x(421), - - x - w - r - - (124).

Проделать то же самое в символьном виде.

Команда **chown** (*CHange OWNer – сменить владельца*) – позволяет сменить владельца файла. Для использования этой команды необходимо либо иметь права владельца текущего файла или права root. Синтаксис команды прост:

```
chown username:groupname filename, где  
username – имя пользователя – нового владельца файла;  
groupname – имя группы – нового владельца файла;  
filename – имя файла, у которого сменяется владелец.
```

Имя группы в синтаксисе команды можно не указывать, тогда будет изменен только владелец файла.

Команда **chgrp** используется для изменения владельца-группы файла. Синтаксис ее таков:

```
chgrp groupname filename,  
где:  
groupname – имя группы, которой будет принадлежать файл  
filename – имя изменяемого файла
```

Имейте в виду, что использовать команды chown и chmod может только пользователь-владелец файла и root, а команду chgrp – пользователь-владелец файла, группа-владелец файла и root.

4. Существуют еще несколько особых прав, которые могут устанавливаться на файлы и каталоги. Например, так называемый sticky bit (бит прикрепления).

В первых версиях Юникс этот бит использовался для того, чтобы заставить систему при работе программы оставлять образ ее кода в памяти. Тогда при следующем обращении к программе на ее запуск тратилось намного меньше времени так как чтение кода с устройства более не требовалось. Для файлов и сегодня в Linux осталось прежнее значение этого бита. А вот для каталогов этот атрибут преобрел новое значение. Если sticky bit установлен на каталог, то удалить файлы из такого каталога может только пользователь-владелец файла, и то только если у него есть право на запись в файл. Группа-владелец и остальные пользователи даже при наличии прав на запись в файл не смогут удалить его при установленном на каталог sticky bit.

Бит прикрепления устанавливается командой chmod в символьном виде:
chmod +t filename

Дополнение

Можно выполнять chmod и chown, пользуясь меню в оболочке mc.

Пример из реальной жизни

На лабораторном практикуме nginx работает от имени пользователя и группы **apache**. У пользователя apache нет прав, чтобы читать и изменять содержимое файлов, которые созданы супер-пользователем root.

Поэтому после каждой модификации рабочего каталога виртуального домена необходимо выполнять

```
chown -R apache:apache /var/www/html/
```

Ключ -R обозначает рекурсивно.