

Лабораторная работа #5

Администрирование системы Linux



РАЗРАБОТАЛ

СЕРГЕЙ СТАНКЕВИЧ (SERHEY STANKEVICH)

ЛАБОРАТОРНАЯ РАБОТА #5

Администрирование системы Linux

Цель работы

Цель. Закрепить на практике основы администрирования системы Linux, изучить атрибуты файлов и права доступа к ним, освоить работу с файлами и каталогами.

Краткие теоретические сведения.

Операционные системы, следующие традициям Unix, являются *многозадачными* и *многопользовательскими*.

Это означает, что компьютером могут одновременно пользоваться несколько человек. Например, если компьютер подключен к локальной сети или к Интернету, удаленные пользователи смогут зайти на него через ssh (secure shell – безопасная командная оболочка) и выполнять операции. Фактически удаленные пользователи могут запускать приложения с графическим интерфейсом и получать изображение на удаленном дисплее. X Window System поддерживает такую возможность изначально.

В далекие времена компьютеры не были «персональными», они были большими и дорогими. Компьютерная система университета, например, состояла из большого центрального компьютера в одном здании и терминалов, разбросанных по всему университетскому городку и соединенных с большим центральным компьютером. Компьютер мог одновременно обслуживать множество пользователей.

Поддержка многопользовательского режима работы внедрена в архитектуру операционной системы. Необходим способ определенной «изоляции» пользователей друг от друга, чтобы:

- действия рядового пользователя не должны приводить к аварийному окончанию работы компьютера,
- ни один пользователь не должен иметь возможность вносить изменения в файлы, принадлежащие другому пользователю.

Атрибуты файла

Для разграничения полномочия пользователей в отношении файлов, в ядре реализована концепция прав доступа, согласно которой каждый файл имеет следующие атрибуты:

- владельца (user, owner);
- группу (group);
- другой пользователь (others, world).

Владелец у файла всегда один. В группе может состоять несколько пользователей. Другой пользователь, это тот, кто не обладает правами суперпользователя (root), не является владельцем и не состоит в указанной группе. В терминологии Unix его еще называют «мир» (world).

Для каждой из перечисленных категорий (владелец, группа, другие) устанавливаются индивидуальные права доступа исходя из трех критериев:

1. Право на чтение (r – read)
2. Право на запись (w – write)
3. Право на выполнение (x – execute)

Команда **ls** – выводит список содержимого каталога. Параметр (флаг) **-l** требует использования «длинного» (long) формата вывода.

```
[me@lbox ~]$ ls -l foo.txt
-rw-rw-r-- 1 me  me  0 2012-03-06 14:52 foo.txt
```

Первые 10 символов в выводе — это *атрибуты прав доступа к файлу*.

1	2	3	4
-	rwX	rw-	r - -

- 1 Тип файла (см. табл. 9.1)
- 2 Привилегии для владельца (см. табл. 9.2)
- 3 Привилегии для группы (см. табл. 9.2)
- 4 Привилегии для всех остальных (см. табл. 9.2)

Типы пользователей

В ОС Linux существует три типа пользователей:

- **root** (корень) – суперпользователь;
- **Системные пользователи** (администраторы);
- **Обычные пользователи.**

Суперпользователь, это аккаунт в UNIX-подобных системах, владелец которого имеет право на выполнение всех операций без исключения. *Присутствует в системе по умолчанию.*

Администраторы – системные процессы, у которых есть учетные записи для управления привилегиями и правами доступа к файлам и каталогам. *Первый администратор создается автоматически при установке системы.*

Обычные пользователи – учетные записи пользователей, допущенных к управлению системой. *Создаются системным администратором.*

Здесь необходимо отметить, что **владельцем** файла или другого ресурса может быть один из трех вышеуказанных пользователей.

В модели безопасности Unix *пользователь* может *владеть* файлами и каталогами. Если пользователь владеет файлом или каталогом, он может управлять доступом к нему.

Пользователи могут также принадлежать *группе*, состоящей из одного или нескольких пользователей, и получить права доступа к файлам и каталогам для членов группы, которые определяются владельцами.

Кроме того, владелец может также определить некоторые права доступа для всех *остальных* (world или others).

Изменение режима доступа к файлу

Права доступа к файлу или каталогу может изменить только владелец.

Команда `chmod` поддерживает два разных способа изменения режима:

- с использованием восьмеричных чисел;
- символическое представление.

Таблица 9.4. Режимы доступа к файлу в двоичном и восьмеричном представлениях

Восьмеричное	Двоичное	Режим доступа
0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r-x
6	110	rw-
7	111	rwX

Определим режим доступа к файлу:

```
[me@lbox ~]$ ls -l foo.txt
-rw-rw-r-- 1 me me 0 2012-03-06 14:52 foo.txt
```

```
[me@lbox ~]$ chmod 600 foo.txt
[me@lbox ~]$ ls -l foo.txt
-rw----- 1 me me 0 2012-03-06 14:52 foo.txt
```

Наиболее часто используются лишь несколько наиболее популярных шаблонов:

7 (rwx)	6 (rw-)	5 (r-x)	4 (r--)	0 (---)
---------	---------	---------	---------	---------

Есть еще одно правило. 4 – читать, 2 – писать, 1 – исполнять.

Добавление пользователя и создание его аккаунта

Добавление аккаунта можно произвести с помощью *интерфейса командной строки* (CLI) или с *графического пользовательского интерфейса* (GUI).

В мире Unix, всегда проводилась четкая грань между обычными пользователями и администраторами. Идеология Unix заключается в том, чтобы предоставлять привилегии суперпользователя, *только когда они действительно необходимы*. Нарушение данной политики ухудшает защищенность Linux, низводя ее до уровня Windows.

Отличие между пользователями **Administrator** и **Standard** лишь в том, что **Administrator** может использовать команду `sudo`.

sudo — выполнение команды от имени другого пользователя

По умолчанию Debian/Ubuntu запрещает регистрироваться в системе с учетной записью root (не позволяя устанавливать пароль для этой учетной записи), а для получения привилегий суперпользователя предлагает использовать sudo.

Начальная учетная запись пользователя обладает полным доступом к привилегиям суперпользователя через sudo и может наделять аналогичными привилегиями другие, вновь создаваемые учетные записи.

Администратор может определить порядок использования sudo обычными пользователями, ограничив возможность запуска команд от имени другого пользователя (обычно суперпользователя). В частности, пользователю может быть разрешен доступ к одним командам и запрещен к другим.

Команда sudo не требует ввода пароля суперпользователя. Для аутентификации в команде sudo пользователь должен ввести *свой пароль (не забывайте свой пароль!)*.

Например, запустить мою программку **my_backup_script_program**, требующую привилегий суперпользователя с помощью sudo можно запустить так:

```
[me@lbox ~]$ sudo my_backup_script_program
Пароль:
System Backup Starting...
```

Чтобы увидеть, какие привилегии дает команда sudo, вызовите ее с параметром **-l** (эль):

```
[me@linuxbox ~]$ sudo -l
User me may run the following commands on this host:
(ALL) ALL
```

Чтобы просто перейти в режим суперпользователя и работать от имени root наберите команду:

```
[me@linuxbox ~]$ sudo -i
```

При этом вы заметите, как измениться знаки привилегии режима работы.

```
irud@ubuntu1:~$ sudo -i
[sudo] password for irud:
root@ubuntu1:~#
root@ubuntu1:~#
root@ubuntu1:~#
root@ubuntu1:~# logout
irud@ubuntu1:~$
```

CTRL +D

Для выхода из привилегированного режима (root) используйте команду **logout** или сочетание клавиш **CTRL-D**.

Как работает команда sudo

При запуске этой команды она читает конфигурационный файл **/etc/sudoers**, в котором храниться список групп пользователей, которым доступны супер-привилегии, то есть группа администраторов. Фрагмент файла представлен:

```
# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:
```

Посмотрим сколько администраторов у нас **/etc/group**.

```
root@DESKTOP-61Q8VTL:/etc# cat group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog, stank
tty:x:5:syslog
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:stank
fax:x:21:
voice:x:22:
cdrom:x:24:stank
floppy:x:25:stank
tape:x:26:
sudo:x:27:stank
audio:x:29:stank
dip:x:30:stank
www-data:x:33:
backup:x:34:
render:x:109:
syslog:x:110:
tss:x:111:
uucidd:x:112:
tcpdump:x:113:
ssh:x:114:
landscape:x:115:
admin:x:116:
netdev:x:117:stank
lxd:x:118:
stank:x:1000:
```

В нашем случае я здесь один. Но список может быть большим, логины администраторов отделяются запятой.

Добавление пользователя посредством CLI

Работать с GUI это не наш метод. Продвинутый *линуксойд* работает с консолью. Для добавления пользователя применяется команда **useradd**.

Добавить пользователя можно определив ему разные свойства. Например, можно создать пользователя, у которого будет своя **home**-директория (используйте флаги **-m**, или **--create-home**), или без нее.

```
[me@lbox ~]$ sudo useradd -m user1
```

Создайте домашний каталог пользователя, если он не существует. Иначе создается пользователь без собственной **home**-директории:

```
[me@lbox ~]$ sudo useradd user1
```

Получение информации о идентичности пользователя

Получить информацию о своей идентичности можно с помощью команды **id**:

```
$ id
uid=500(me) gid=500(me) groups=500(me)
```

Пример взят из системы Fedora. Когда создается учетная запись пользователя, ей присваивается число, которое называют идентификатором пользователя (user ID), или **uid**. Это число, исключительно ради удобства человека, отображается как имя пользователя. Пользователю назначается идентификатор основной группы (primary group ID), или **gid**, и дополнительно пользователь может включаться в состав других групп.

В других системах, таких как Ubuntu, вывод команды может немного отличаться. В дистрибутиве Fedora нумерация учетных записей обычных пользователей начинается с 500, тогда как в Ubuntu – с 1000.

Кроме того, пользователь в Ubuntu принадлежит множеству других групп. Это связано с особенностями управления привилегиями доступа к системным устройствам и службам в Ubuntu.

```
$ id
uid=1000(me) gid=1000(me)
groups=4(adm), 20(dialout), 24(cdrom), 25(floppy), 29(audio), 30(dip),
44(video), 46(plugdev), 108(lpadmin), 114(admin), 1000(me)
```

А где же вся эта информация хранится?

Специальные конфигурационные файлы пользователей

Как и многое другое в Linux, она хранится в паре текстовых, конфигурационных файлов. Учетные записи пользователей хранятся в файле `/etc/passwd`, а информация о группах – в файле `/etc/group`. Когда создаются новые учетные записи и группы, эти файлы изменяются вместе с файлом `/etc/shadow`, где хранится информация о пароле пользователя.

Для каждой учетной записи в файле `/etc/passwd` определяется имя пользователя (для входа), числовой идентификатор пользователя (`uid`), числовой идентификатор основной группы (`gid`), действительное имя пользователя, путь к домашнему каталогу и командная оболочка входа (`login shell`).

Итак, вся информация о пользователях (аккаунтах) хранится в файле **`/etc/passwd`**. Это обычный текстовый файл, право на чтение которого имеют все пользователи системы, а право на запись имеет только администратор (суперпользователь). Добавление пользователя в систему сводится к внесению в файл `/etc/passwd` соответствующей записи.

Каждая строка файла `passwd` является записью конкретного пользователя и имеет формат – *семь* полей (атрибутов), разделенных двоеточиями.

Заглянув внутрь `/etc/passwd` и `/etc/group`, можно заметить, что помимо учетных записей обычных пользователей здесь также хранятся учетные записи суперпользователя (`uid 0`) и различных других системных пользователей. Изучая тему о процессах, вы узнаете, что некоторые из этих других «пользователей» в действительности существуют не просто так. Рассмотрим фрагмент файла `/etc/passwd`:

```
root:x:0:0:root:/:bin/bash
daemon:x:1:1:daemon:/:
bin:x:2:2:bin:/:usr/bin:
sys:x:3:3:sys:/:
adm:x:4:4:adm:/:var/adm
lp:x:71:8:lp:/:usr/spool/lp:
uucp:x:5:5:uucp:/:usr/lib/uucp:
nobody:x:60001:60001:nobody:/:
andy:x:206:101:Andrei Robachevsky:/home/andy:/bin/bash
```

name:passwd-encod:UID:GID:comments:home-dir:shell

В системе информация о пользователе хранится также в других местах. Поэтому создание пользователя простым редактированием файла `/etc/passwd` может привести к неправильной регистрации пользователя, или к нарушениям работы системы. Вместо этого следует пользоваться специальными утилитами, поставляемыми с системой.

Однако, если пользователь установлен в системе, но у него нет пароля, это означает, что для этого пользователя нет аккаунта, т.е. он не авторизован. Для установки пароля, т.е. создания аккаунта, а также для изменения пароля пользователя применяется команда **passwd**.

Команда **passwd** имеет следующий синтаксис:

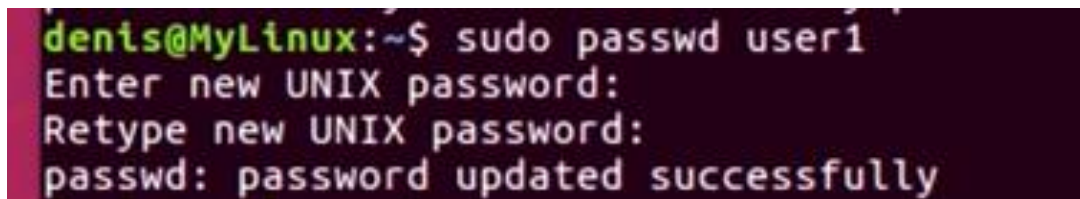
```
passwd [ -[опции]] [пользователь]
```

Опции команды рассмотрим несколько позже.

Изменение своего пароля:

```
me@lbox ~]$ passwd
Смена пароля для me.
Введите новый пароль UNIX:
Подтвердите новый пароль UNIX:
passwd: пароль обновлен успешно.
```

Изменение пароля пользователя:



```
denis@MyLinux:~$ sudo passwd user1
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

В файле `/etc/passwd` хранятся пароли пользователей, в зашифрованном виде. Открытость паролей в файле может стать недостатком с точки зрения безопасности, поэтому во многих системах зашифрованные пароли хранятся в отдельном закрытом для чтения и записи файле `/etc/shadow`. На наличие зашифрованного пароля, который храниться в другом месте, указывает 2-й атрибут – «х». Чтобы посмотреть файл паролей используем команду `shadow` с суперпривилегиями.

Заглянем в файл `/etc/shadow`. Например, так:

```
[me@lbox ~]$ sudo cat /etc/shadow
```

```
denis@MyLinux:/home$ cat /etc/shadow
cat: /etc/shadow: Permission denied
denis@MyLinux:/home$ sudo cat /etc/shadow
root:!:17092:0:99999:7:::
daemon*:17001:0:99999:7:::
bin*:17001:0:99999:7:::
sys*:17001:0:99999:7:::
sync*:17001:0:99999:7:::
...
saned*:17001:0:99999:7:::
usbmux*:17001:0:99999:7:::
denis:$6$bQ0hpk4p$yUUTuKh.V/8B3d7vAP9Rz6BvkMQuQ70QzsmRUbwvI3geKoF2oaX0/Gn0DBEl6lqoxQLi70UVAdpcV9Dyky
gal.:17092:0:99999:7:::
vboxadd!:17092:::
vasya:$6$4EjrSd90$g0lhPLG/97tXgJSpGPlapMAdAEVZtQimIAAEFPDDLeaUTgiTz.l.rTNIkm6JC7kJJbMVw8NyScy.UTU3rr
3rR/:17125:0:99999:7:::
petya:$6$0oTBgcEJ$51ViMv/mt3sWZ3bBbXLHQE8wkDCQ3x7.2H2CUVf0SIcmbV6bwTONYo5oRv.3etnwAb02Sv9VxA2Di2/QsR
DBW0:17125:0:99999:7:::
koilya!:17125:0:99999:7:::
denis@MyLinux:/home$
```

Атрибуты пароля пользователя (файл **/etc/shadow**):

1. логин;
2. зашифрованный пароль;
3. дата последней смены пароля;
4. минимальный возраст пароля (сутки);
5. максимальный возраст пароля (сутки);
6. период предупреждения пароля;
7. период бездействия пароля;
8. дата истечения срока действия аккаунта;
9. зарезервированное поле.

Здесь под датой имеется ввиду количество дней от наступления эры *UNIX*.

Особое внимание уделим второму атрибуту – зашифрованный пароль. Параметром атрибута может быть огромный шифр пароля, или символ « ! », который означает отсутствие пароля реального пользователя, логин не создан. Также может присутствовать знак « * », *у него нет пароля, см. Робачевский и лекции Администрирование.*

Первая цифра обозначает форматы шифрования паролей. Бывают следующие форматы:

```
$1$ is MD5
$2a$ is Blowfish
$2y$ is Blowfish
$5$ is SHA-256
$6$ is SHA-512
```

Внесение изменений значений вышеуказанных атрибутов суперпользователю доступна возможность блокировки учетных записей, установки времени действия пароля и многое другое. Напоминаем, вносить изменения следует специальными утилитами.

Вернемся к файлу **/etc/passwd**. Если пользователь зарегистрирован в системе и ввел правильный пароль, запускает программу, указанную в 7-м поле записи пользователя в файле В принципе это может быть любая программа, но в нашем случае – это командный интерпретатор **bash (shell)**.

Как заставить пользователя поменять пароль

Безопасность сервера – это одна из самых важных задач администрирования. Часто причиной проблем с безопасностью становятся сами пользователи, которые недостаточно часто меняют пароли или делают их слишком простыми. Если вы администратор, у вас есть возможность заставить пользователей выполнять смену пароля время от времени, а также автоматически отсылать им предупреждения о том, что пора сменить пароль пользователя Linux.

Всё это позволяет сделать утилита **passwd** с использованием опций (флагов):

- d** - удалить пароль пользователя, после этого он не сможет войти
- e** - сделать пароль устаревшим
- i** - через сколько дней после того, как пароль устарел, отключить аккаунт, если пользователь не сменил пароль
- l** - запретить пользователю входить в систему
- n** - минимальное количество дней между сменами пароля
- S** - отобразить информацию об аккаунте
- u** - отменяет действие параметра **-l**
- x** - максимальное количество дней, пока пароль можно использовать.
- w** - количество дней, после которых нужно предупреждать пользователя о том, что надо сменить пароль.

Изменение идентичности

Время от времени возникает необходимость приобрести идентичность другого пользователя. Чаще всего требуется получить привилегии суперпользователя, чтобы выполнить некоторые административные задачи. Так же можно «превратиться» в другого обычного пользователя, чтобы, к примеру, проверить настройки учетной записи.

Существует три способа приобрести альтернативную идентичность:

- выйти из системы и войти вновь с учетными данными другого пользователя (это долго и неудобно);
- воспользоваться командой `su` (удобно и быстро);
- воспользоваться командой `sudo` (удобно и быстро).

`su` — запуск командной оболочки с подстановкой идентификаторов пользователя и группы

Команда **`su`** используется для запуска нового сеанса работы с командной оболочкой от имени другого пользователя. У начинающих пользователей бытует заблуждение, они трактуют команду как «~~super-user~~», но это не так. Команду следует понимать как «switch user» — выбрать пользователя.

Команда имеет следующий синтаксис:

`su [-[флаг-параметр]] [пользователь]`

Запустить командную оболочку от имени суперпользователя можно так:

```
[me@lbox ~]$ su           тоже самое           [me@lbox ~]$ su root
Password:
[root@lbox ~]#
```

После ввода правильного пароля появится новое приглашение к вводу, показывающее, что данная командная оболочка обладает привилегиями суперпользователя (символ `#` в конце вместо символа `$`) и текущим рабочим каталогом теперь стал домашний каталог суперпользователя (обычно `/root`).

Несмотря на получение прав `root`, вы остаетесь в своей пользовательской домашней директории

```
[me@lbox ~]# pwd
/home/me
```

То есть вы остались тем же пользователем `me`, только наделенным правами суперпользователя. Поэтому не удивляйтесь, что некоторые директории и команды по-прежнему будут для вас закрыты (эти директории и команды определяются политикой безопасности вашего дистрибутива). Дело в том, что вы сохранили не только свою домашнюю директорию, но и свои переменные окружения (`environmental variables`), которые вовсе не совпадают с таковыми суперпользователя. Правда, зная в какой директории находятся исполняемые

файлы нужных команд, легко обойти этот запрет, прописав полный путь вручную. Кажется это ограничение только для домохозяек и домохозяев.

Однако команда `su` дает возможность стать полноценным суперпользователем, не только получив его права, но и перейдя в его домашнюю директорию. Фактически вы, не прерывая сессии, перелогиниваетесь в `root`. Для этого необходимо после команды `su` оставить пробел, а затем ввести символ черты (-):

```
[me@lbox ~]$ su -           тоже самое      [me@lbox ~]$ su -l (--login)
Password:
[root@lbox ~]#
[me@lbox ~]# pwd
/root
```

Вместе с домашней директорией суперпользователя, вы получаете и его переменные окружения, в том числе и пути.

Команда `su` и рядовые пользователи

Введя после `su` аргумент (логин другого пользователя), вы можете получить его права. Естественно, если знаете его пароль.

Например, моей жене срочно понадобилось посмотреть свою почту, я набираю:

```
[ya@antony ~]$ su - wife      (wife – это логин жены)
```

Затем я отворачиваюсь, и жена вводит свой пароль (*правила безопасности*: не знайте пароль жены, и никогда не читайте ее почту):

```
Password:
[wife@antony ~]$
```

открывает от своего "имени" свой почтовый клиент, просматривает свои письма, и, поблагодарив, набирает `exit` и уходит:

```
[wife@antony ~]$ exit
Password:
[ya@antony ~]$
```

И не пришлось закрывать программы, перелогиниваться и прочее. Удобно. А вводить логин жены через черточку пришлось, чтобы она попала в ее домашнюю директорию, где находится ее почтовый клиент.

Удаление учетной записи пользователя из системы

Это делается с помощью команды `userdel`. Смотри справочник `man`.

Правила безопасности

На многопользовательских системах, не только работа под аккаунтом суперпользователя, но и применение команды `su` должно быть сведено к необходимому минимуму, и применяться с осторожностью. Следуйте правилам:

1. Посторонние (не доверенные) лица не должны видеть никаких паролей, вводимых после команды `su`.
2. Не оставлять без присмотра машины с открытым сеансом `su`.
3. По окончании работы в сеансе `su` немедленно закрывать его.

Чтобы закончить работу и вернуться в предыдущую командную оболочку введите команду **exit**:

```
[root@lbox ~]# exit
[me@lbox ~]$
```

Отличия команд `sudo` и `su`.

Команда `sudo` не требует ввода пароля суперпользователя. Для аутентификации в команде `sudo` пользователь должен ввести свой пароль.

Также `sudo` — не запускает новую командную оболочку и не загружает окружение другого пользователя.

Создание и удаление группы владельцев файла

Несмотря на то, что во многих Unix-подобных системах обычных пользователей включают в общую группу, такую как `users`, в современных дистрибутивах Linux принято создавать для каждого пользователя свою, уникальную группу с одним членом и именем, совпадающим с именем пользователя. Это упрощает распределение определенных типов привилегий.

```
denis@MyLinux:~$ sudo groupadd Programmers
denis@MyLinux:~$ sudo groupadd Marketing
denis@MyLinux:~$ cat /etc/group|
```

И в содержимом файла `/etc/group` появятся данные о новых группах:

```
kotya:x:1003:
user2:x:1005:
Programmers:x:1006:
Marketing:x:1007:
denis@MyLinux:~$
```

Чтобы удалить группу используйте команду и посмотрите содержимое файла `/etc/group`:

```
denis@MyLinux:~$ sudo groupdel testers
denis@MyLinux:~$ cat /etc/group
```

Изменение владельца и группы файла

Команда **chown** используется для изменения владельца и группы файла или каталога. Для использования этой команды необходимы *привилегии супер-пользователя*. Команда имеет следующий синтаксис:

`chown [владелец][:[группа]] файл...`

Команда может изменить владельца и/или группу файла в зависимости от первого аргумента. В таблице приводятся несколько примеров команды.

Таблица 9.7. Примеры аргументов команды `chown`

Аргумент	Результаты
<code>bob</code>	Изменит принадлежность файла, назначив владельцем пользователя <code>bob</code>
<code>bob:users</code>	Изменит принадлежность файла, назначив владельцем пользователя <code>bob</code> и группу <code>users</code>
<code>:admins</code>	Изменит принадлежность файла, назначив группу <code>admins</code>
<code>bob:</code>	Изменит принадлежность файла, назначив владельцем пользователя <code>bob</code> и группу этого пользователя

Изменение группы файла

В старых версиях Unix команда `chown` изменяла только владельца файла, но не группу. Чтобы изменить группу, предоставлялась другая команда, `chgrp`. Она действует практически так же, как `chown`, но имеет больше ограничений.

Добавление и удаление пользователя в группу

```
denis@MyLinux:~$ id vasya
uid=1001(vasya) gid=1001(vasya) groups=1001(vasya)
denis@MyLinux:~$ sudo usermod -aG Marketing vasya
denis@MyLinux:~$ id vasya
uid=1001(vasya) gid=1001(vasya) groups=1001(vasya),1007(Marketing)
denis@MyLinux:~$ sudo usermod -aG sudo vasya
denis@MyLinux:~$ id vasya
uid=1001(vasya) gid=1001(vasya) groups=1001(vasya),27(sudo),1007(Marketing)
```


Чтобы удалить пользователя из группы продолжим так:

```
denis@MyLinux:~$ sudo deluser vasya Marketing
Removing user `vasya' from group `Marketing' ...
Done.
denis@MyLinux:~$ id vasya
uid=1001(vasya) gid=1001(vasya) groups=1001(vasya),27(sudo) I
denis@MyLinux:~$
```

Липкий бит (Sticky bit)

Администратор может запретить другим пользователям запускать, удалять и проводить другие операции с файлами, установив нужные права доступа файлов. Однако, что делать, когда в системе два и более администратора? Каждый администратор может действовать по своему усмотрению. Чтобы разрешить этот конфликт можно использовать **липкий бит**.

Предоставление права на запись в каталог дает достаточно большие полномочия. Имея такое право, пользователь может удалить из каталога любой файл, даже тот, владельцем которого он не является и в отношении которого не имеет никаких прав.

Установка атрибута Sticky bit для каталога позволяет установить дополнительную защиту файлов, находящихся в каталоге. Из такого каталога пользователь может удалить только файлы, которыми он владеет, или на которые он имеет явное право доступа на запись, даже при наличии права на запись в каталог.

Примером может служить каталог /tmp, который является открытым на запись для всех пользователей, но в котором может оказаться нежелательной возможность удаления пользователем чужих временных файлов.

Жесткие и символические ссылки

Ссылки на файлы в файловой системе Linux бывают двух типов:

- символические ссылки (symbolic links);
- жесткие (прямые) ссылки (hard links).

Пользователь может создавать ссылки при помощи программы ln.

ln файл ссылка

создает жесткую ссылку.

ln -s элемент ссылка

создает символическую ссылку, где элементом может быть файл или каталог

Жесткие ссылки – это первоначальный способ создания ссылок в Unix; символические ссылки – более позднее изобретение. По умолчанию каждый файл имеет одну жесткую ссылку, определяющую его имя. Создавая жесткую ссылку, мы создаем дополнительную запись в каталоге для файла. Жесткие ссылки имеют два важных ограничения:

- Жесткая ссылка не может указывать на файл за пределами собственной файловой системы. Это означает, что ссылка не может указывать на файл, находящийся в другом разделе диска.
- Жесткая ссылка не может указывать на каталог.

Жесткая ссылка неотличима от самого файла. В отличие от символических ссылок, при выводе списка с содержимым каталогов жесткие ссылки никак не выделяются. При удалении жесткой ссылки удаляется только сама ссылка, а файл остается на месте (то есть пространство, занимаемое файлом, не освобождается), пока не будут удалены все жесткие ссылки на файл.

Рассуждая о жестких ссылках, полезно представлять файлы состоящими из двух частей: раздела с данными, где хранится содержимое файла, и раздела с именем, где хранится имя файла. Создавая жесткую ссылку, мы фактически создаем дополнительный раздел с именем, ссылающийся на тот же раздел с данными. Цепочку дисковых блоков система присваивает тому, что называется индексным узлом (inode), который затем присваивается разделу с именем. То есть каждая жесткая ссылка ссылается на определенный индексный узел с содержимым файла.

Команда `ls` может извлекать эту информацию. Для этого ее нужно вызвать с параметром `-li`:

```
$ ls -li
```

Символические ссылки были придуманы с целью преодолеть ограничения жестких ссылок. Когда создается символическая ссылка, в действительности создается файл особого типа, содержащий текстовый указатель на файл или каталог. В некотором отношении они действуют подобно ярлыкам в Windows.

Файл, на который указывает символическая ссылка, и сама символическая ссылка почти неотличимы друг от друга. Например, если попытаться что-то записать в символическую ссылку, запись будет выполнена в файл, на который она указывает. Однако при удалении символической ссылки удаляется только символическая ссылка, но не файл. Если удалить файл до того, как будет удалена символическая ссылка, ссылка останется на месте, но будет указывать в никуда. О таких ссылках говорят, что они «битые». Во многих реализациях

команда `ls` выделяет битые ссылки цветом, например, красным, чтобы обратить на них внимание.

Вспомогательны команды

Для проверки идентичности:

id [username]— выводит информацию об идентичности пользователя, показывает к каким группам принадлежит пользователь;

whoami — показать им текущего пользователя;

who — показать сколько пользователей сейчас в системе;

w — показать кто сейчас в системе и что делает (также как **who**, но меньше букв, но больше информации);

last — показать последние логины;

для установки шаблонов:

umask — определяет разрешения доступа к файлам по умолчанию;

скелет — фа фа

Таблица 1.4. Дополнительные атрибуты для каталогов [3]

Код	Название	Значение
t	Sticky bit	Позволяет пользователю удалять только файлы, которыми он владеет или имеет права на запись
s	Set GID, SGID	Позволяет изменить правило установки владельца-группы создаваемых файлов, аналогично реализованному в BSD UNIX

Для более подробной и достоверной информации используйте 8-й и 5-й разделы справочника **man**.

```
4 Special files (usually found in /dev)
5 File formats and conventions, e.g. /etc/passwd
6 Games
```

```
7 Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7)
8 System administration commands (usually only for root)
9 Kernel routines [Non standard]
```

УПРАЖНЕНИЯ

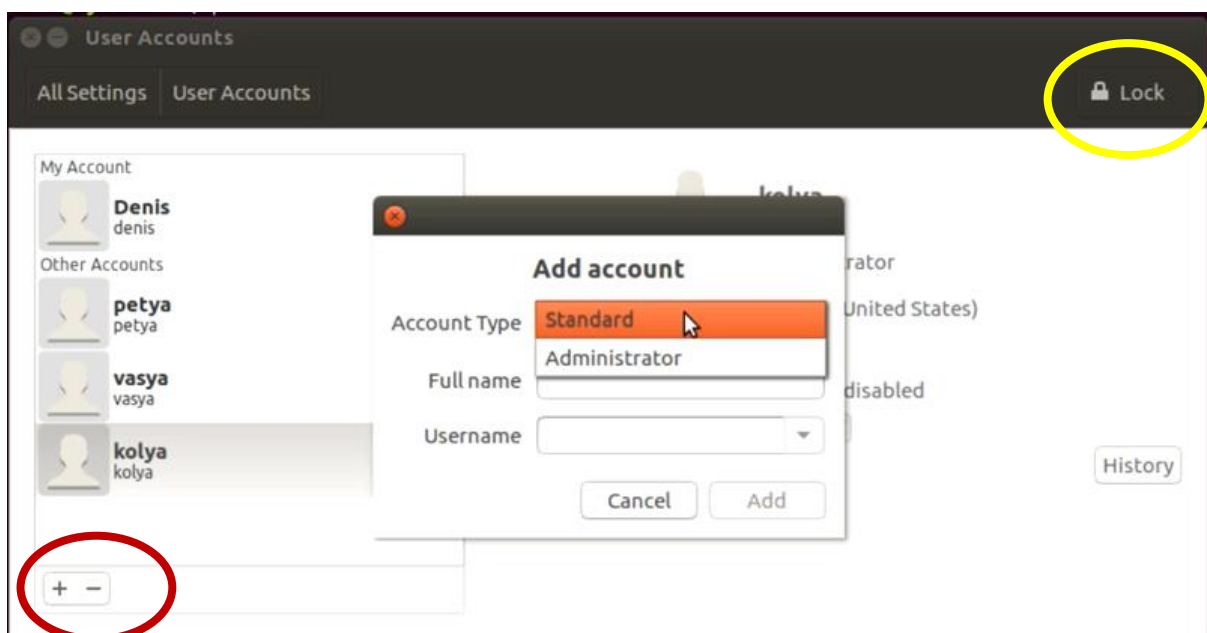
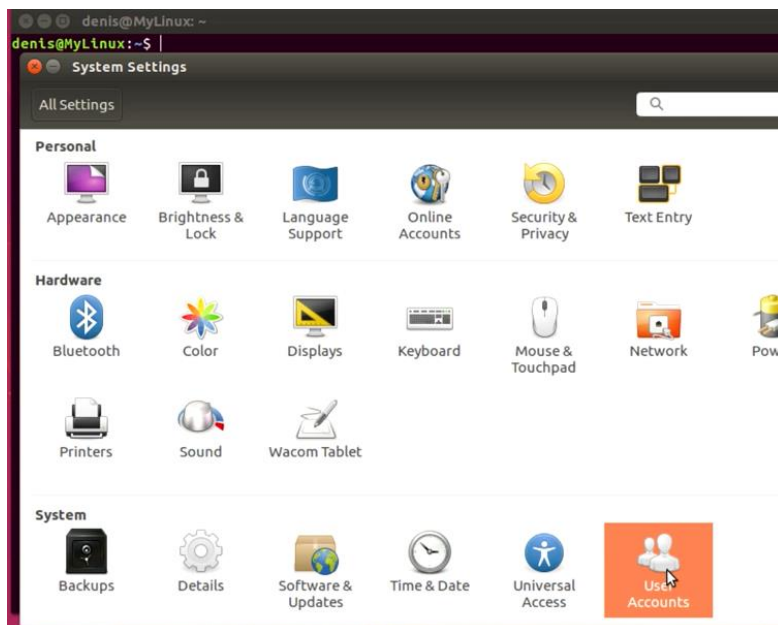
Упражнение 1

В этом упражнении вы научитесь создавать и удалять учетные записи пользователей системы, изменять *настройки пароля* стандартного пользователя.

Создайте *стандартного* пользователя с помощью *GUI* и *CLI*.



Добавление пользователя посредством GUI:



Добавьте одного стандартного пользователя с помощью графической утилиты.

Перейдите в директорию /home и посмотрите какие учетные записи там есть.

```
denis@MyLinux: /home
denis@MyLinux:~$ cd /home
denis@MyLinux:/home$ ls -l
total 16
drwxr-xr-x 21 denis denis 4096 Nov 20 12:26 denis
drwxr-xr-x  2 kolya kolya 4096 Nov 20 12:37 kolya
drwxr-xr-x  3 petya petya 4096 Nov 20 12:42 petya
drwxr-xr-x  3 vasya vasya 4096 Nov 20 12:33 vasya
denis@MyLinux:/home$ |
```

Добавьте пользователя с помощью консольной команды **useradd**:

```
denis@MyLinux:~$ sudo useradd -m user1
denis@MyLinux:~$ ls -l /home
total 20
drwxr-xr-x 21 denis denis 4096 Nov 20 12:26 denis
drwxr-xr-x  2 kolya kolya 4096 Nov 20 12:37 kolya
drwxr-xr-x  3 petya petya 4096 Nov 20 12:42 petya
drwxr-xr-x  2 user1 user1 4096 Nov 20 13:53 user1
drwxr-xr-x  3 vasya vasya 4096 Nov 20 12:33 vasya
denis@MyLinux:~$ |
```

Зайдите в директорию нового пользователя и посмотрите, что там есть.

При создании учетной записи пользователя имеется интересная возможность автоматически создавать в домашней директории этого пользователя нужные нам файлы и другие директории. В системе имеется специальная директория /etc/skel, в которую мы можем помещать нужные нам файлы шаблоны. При создании нового пользователя содержимое директории каркаса (скелета) копируется в домашнюю директорию вновь создаваемого пользователя. Директория /etc/skel обычно содержит скрытые файлы .bashrc.

Прделаем следующие действия:

```
denis@MyLinux:/home/user1$ cd /etc/skel/
denis@MyLinux:/etc/skel$ ls -l
total 12
-rw-r--r-- 1 root root 8980 Apr 20 2016 examples.desktop
denis@MyLinux:/etc/skel$ mkdir Desktop
mkdir: cannot create directory 'Desktop': Permission denied
denis@MyLinux:/etc/skel$ sudo mkdir Desktop
denis@MyLinux:/etc/skel$ sudo mkdir Video
denis@MyLinux:/etc/skel$ touch myfile.txt
touch: cannot touch 'myfile.txt': Permission denied
denis@MyLinux:/etc/skel$ sudo touch myfile.txt
denis@MyLinux:/etc/skel$ ls -l
total 20
drwxr-xr-x 2 root root 4096 Nov 20 13:56 Desktop
-rw-r--r-- 1 root root 8980 Apr 20 2016 examples.desktop
-rw-r--r-- 1 root root  0 Nov 20 13:56 myfile.txt
drwxr-xr-x 2 root root 4096 Nov 20 13:56 Video
denis@MyLinux:/etc/skel$ |
```


Затем создадим нового пользователя `$ sudo useradd -m user2`, и получим:

```
denis@MyLinux:~$ ls -l /home/user2
total 20
drwxr-xr-x 2 user2 user2 4096 Nov 20 13:56 Desktop
-rw-r--r-- 1 user2 user2 8980 Apr 20 2016 examples.desktop
-rw-r--r-- 1 user2 user2 0 Nov 20 13:56 myfile.txt
drwxr-xr-x 2 user2 user2 4096 Nov 20 13:56 Video
denis@MyLinux:~$
```

Содержимое директории `skel` скопировалось в домашнюю директорию нового пользователя.

Изменение пароля

У нового пользователя появилась своя директория, однако у него отсутствует пароль для входа в систему и логин. В этом можно убедиться, получив информацию из файла **shadow**.

Откройте метаданные всех пользователей в нашей системе, которые хранятся в файле **passwd**.

```
denis@MyLinux:/home$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
```

Найдите здесь метаданные всех зарегистрированных пользователей системы:

```
usbmux:x:120:40:usbmux daemon,,,:/var/ctb/usbmux:/bin/false
denis:x:1000:1000:Denis,,,:/home/denis:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
vasya:x:1001:1001::/home/vasya:
petya:x:1002:1002::/home/petya:
kolya:x:1003:1003:kolya,,,:/home/kolya:/bin/bash
denis@MyLinux:/home$
```

Поясните, что обозначают эти данные.

Создание, изменение и настройка паролей пользователей

Откройте содержимое файла **shadow**, предварительно получив суперпривилегии:

```
denis@MyLinux:/home$ cat /etc/shadow
cat: /etc/shadow: Permission denied
denis@MyLinux:/home$ sudo cat /etc/shadow
```

Найдите атрибуты паролей учетных записей пользователей вашей системы, и дайте пояснения, что они обозначают.

```
usbmux:*:17001:0:99999:7:::
denis:$6$bQ0hpk4p$yUUTuKh.V/8B3d7vAP9Rz6BvkMQuQ70QzsmRUBwvI3geKoF2oaX0/Gn0DBEl6lqoxQLi70UVApcV9Dyky
gal.:17092:0:99999:7:::
vboxadd!:17092:0:99999:7:::
vasya:$6$4EjrSd90$g0lhPLG/97tXgJSpGPlapMAdAEVZtQinIAAEFPDDLeaUTgiTz.l.rTNIk6JC7kJJbMVw8NyScy.UTU3rr
3rR/:17125:0:99999:7:::
petya:$6$0oTBgcEJ$51ViMv/mt3sWZ3bBbXLHQE8wkDCQ3x7.2H2CUVf05IcmbV6bwT0NYo5oRv.3etnwAb025v9VxA2Dl2/QsR
DBW0:17125:0:99999:7:::
kolya!:17125:0:99999:7:::
user1!:17125:0:99999:7:::
denis@MyLinux:~$ passwd user1
```

Если у пользователя, уже зарегистрированного в системе отсутствует пароль, то пароль можно установить следующим образом:

```
denis@MyLinux:~$ sudo passwd user1
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
denis@MyLinux:~$ |
```

Точно таким же способом можно изменить существующий пароль.

Попробуем изменить настройки пароля стандартного пользователя. Сначала посмотрим информацию о пароле: `$ sudo passwd -S user`

```
x - fish /home/sergly
sergiy@sergiy-VirtualBox-> sudo passwd -S test
test P 07/21/2016 0 99999 7 -1
sergiy@sergiy-VirtualBox-> |
```

1. Первое поле - имя пользователя
2. Второе поле показывает одно из значений: P - пароль установлен, L - пользователь заблокирован, NP - пароля нет.
3. 07/21/2016 - дата последнего изменения пароля.
4. 0 - минимальное время до смены пароля
5. 99999 - максимальное время действия пароля
6. 7 - за сколько дней нужно предупреждать об истечении срока действия пароля
7. -1 - через сколько дней пароль нужно деактивировать.

Сделаем, чтобы через тридцать дней после смены, пароль пользователя станет устаревшим: `$ sudo passwd -x 30 test`

```
x - □ fish /home/sergly
sergry@sergry-VirtualBox-> sudo passwd -x 1 test
passwd: информация об истечении срока действия пароля изменена.
sergry@sergry-VirtualBox->
```

Внесем другие изменения.

За три дня до того, как пароль устареет, предупредим пользователя, что его нужно сменить:

```
$ sudo passwd -w 3 test
```

Если он этого не сделает в течении пяти дней, аккаунт нужно отключить:

```
$ sudo passwd -i 5 test
```

Пароль можно менять не чаще, чем раз в 10 дней:

```
$ sudo passwd -n 10 test
```

Посмотрим, что у нас получилось:

```
x - □ fish /home/sergly
sergry@sergry-VirtualBox-> sudo passwd -S test
test P 07/21/2016 10 30 3 5
sergry@sergry-VirtualBox->
```

Удаление пользователей

Если вам надоели другие пользователи системы, вы можете удалить их учетные записи.

```
denis@MyLinux:~$ sudo userdel -r user1
userdel: user1 mail spool (/var/mail/user1) not found
denis@MyLinux:~$ ls -l /home
total 20
drwxr-xr-x 21 denis denis 4096 Nov 20 12:26 denis
drwxr-xr-x  2 kolya kolya 4096 Nov 20 12:37 kolya
drwxr-xr-x  3 petya petya 4096 Nov 20 12:42 petya
drwxr-xr-x  4 user2 user2 4096 Nov 20 13:57 user2
drwxr-xr-x  3 vasya vasya 4096 Nov 20 12:33 vasya
denis@MyLinux:~$ cat |
```

И если заглянуть в файл `passwd` наш пользователь исчез и оттуда.

Упражнение 2

В этом упражнении мы научимся создавать и удалять группы пользователей, идентифицировать пользователей, добавлять пользователей в группы и удалять их из групп, а также получать о них нужную информацию.

Создание группы

Выведем содержимое файла **group**, чтобы увидеть метаданные о всех группах в системе: `$ cat /etc/group`

```
denis@MyLinux:/home$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,denis,kolya
tty:x:5:
```

Здесь можно определить какие группы существуют в системе, и какие участники принадлежат конкретной группе, например, кто является администратором системы. Обратите внимание на следующие группы:

```
lcopy:x:25:
tape:x:26:
sudo:x:27:denis,kolya
audio:x:29:pulse
dis:x:30:denis
```

```
saned:x:127:
denis:x:1000:
sambashare:x:128:denis,kolya
vboxsf:x:999:
vasya:x:1001:
petya:x:1002:
kolya:x:1003:
denis@MyLinux:/home$
```

А какие группы есть в вашей системе?

Идентификация пользователя

Для идентификации пользователя используются следующие команды: `whoami`, `id` и другие. Перейдите в сеанс другого пользователя и идентифицируйте его.

```
denis@MyLinux:/home$ whoami
denis
denis@MyLinux:/home$ su vasya
Password:
vasya@MyLinux:/home$ whoami
vasya
vasya@MyLinux:/home$ exit
exit
```

```
denis@MyLinux:/home$ id
uid=1000(denis) gid=1000(denis) groups=1000(denis),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
denis@MyLinux:/home$ id vasya
uid=1001(vasya) gid=1001(vasya) groups=1001(vasya)
denis@MyLinux:/home$ id kolya
uid=1003(kolya) gid=1003(kolya) groups=1003(kolya),4(adm),27(sudo),113(lpadmin),128(sambashare)
denis@MyLinux:/home$ |
```

Состояние активных пользователей системы можно определить с помощью команд **who** и **w**. Запустите терминалы в сеансах всех зарегистрированных в вашей системе пользователей и определите их состояние.

```
denis@MyLinux:/home$ who
denis    tty7          2016-11-20 12:25 (:0)
vasya    tty4            2016-11-20 12:33
petya    tty6            2016-11-20 12:42
denis@MyLinux:/home$ w
 12:56:26 up 31 min,  3 users,  load average: 0.07, 0.09, 0.09
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
denis     tty7      :0              12:25    31:23  32.93s  0.10s  /sbin/upstart --user
vasya     tty4            I        12:33    18.00s  0.02s  0.02s  -bash
petya     tty6            12:42    26.00s  0.02s  0.00s  ping 127.0.0.1
denis@MyLinux:/home$ |
```

Командой **last** проверьте историю сеансов пользователей.

Изменение владельца файла

Создайте директорию, в которую скопируйте какие-нибудь файлы и директории разных владельцев:

```
denis@MyLinux:~/Lesson16$ ls -l
total 4
-rw-rw-r-- 1 denis denis    0 Nov 22 22:00 denis.txt
-rw-rw-r-- 1 vasya vasya    0 Nov 22 21:50 vasya.txt
drwxrwxr-x 2 denis denis 4096 Nov 22 22:15 zzz
denis@MyLinux:~/Lesson16$
```

Измените владельца какого ни будь файла или директории:

```
denis@MyLinux:~/Lesson16$ sudo chown petya zzz/
denis@MyLinux:~/Lesson16$ ls -l
total 4
-rw-rw-r-- 1 denis denis    0 Nov 22 22:00 denis.txt
-rw-rw-r-- 1 vasya vasya    0 Nov 22 21:50 vasya.txt
drwxrwxr-x 2 petya denis 4096 Nov 22 22:15 zzz
denis@MyLinux:~/Lesson16$ |
```

Дайте пояснения по атрибутам доступа к файлам и директориям, то есть какие пользователи и какой доступ имеют к файлам.

Изменение группы владельца файла

```
denis@MyLinux:~/Lesson16$ sudo chgrp Programmers vasya.txt
denis@MyLinux:~/Lesson16$ ls -l
total 4
-rw-rw-r-- 1 denis denis    0 Nov 22 22:00 denis.txt
-rw-rw-r-- 1 vasya Programmers 0 Nov 22 21:50 vasya.txt
drwxrwxr-x 2 petya denis 4096 Nov 22 22:15 zzz
denis@MyLinux:~/Lesson16$ |
```

Упражнение 3

В этом упражнении мы научимся изменять права доступа к файлу. Также научимся создавать ссылки на файлы и работать с ними.

Изменение прав доступа к файлу буквенным способом

```
denis@MyLinux:~/Lesson16$ chmod o+x denis.txt
denis@MyLinux:~/Lesson16$ ls -l
total 4
-rw-rw-r-x 1 denis denis      0 Nov 22 22:00 denis.txt
-rw-rw-r-- 1 vasya Programmers 0 Nov 22 21:50 vasya.txt
drwxrwxr-x 2 petya denis    4096 Nov 22 22:15 zzz
denis@MyLinux:~/Lesson16$
```

```
denis@MyLinux:~/Lesson16$ chmod g-w denis.txt
denis@MyLinux:~/Lesson16$ ls -l
total 4
-rw-r--r-x 1 denis denis      0 Nov 22 22:00 denis.txt
-rw-rw-r-- 1 vasya Programmers 0 Nov 22 21:50 vasya.txt
drwxrwxr-x 2 petya denis    4096 Nov 22 22:15 zzz
denis@MyLinux:~/Lesson16$
```

```
denis@MyLinux:~/Lesson16$ chmod g-w,o+w denis.txt
denis@MyLinux:~/Lesson16$ ls -l
total 4
-rw-r--rwx 1 denis denis      0 Nov 22 22:00 denis.txt
-rw-rw-r-- 1 vasya Programmers 0 Nov 22 21:50 vasya.txt
drwxrwxr-x 2 petya denis    4096 Nov 22 22:15 zzz
denis@MyLinux:~/Lesson16$
```

```
denis@MyLinux:~/Lesson16$ chmod ugo=r denis.txt
denis@MyLinux:~/Lesson16$ ls -l
total 4
-r--r--r-- 1 denis denis      0 Nov 22 22:00 denis.txt
-rw-rw-r-- 1 vasya Programmers 0 Nov 22 21:50 vasya.txt
drwxrwxr-x 2 petya denis    4096 Nov 22 22:15 zzz
denis@MyLinux:~/Lesson16$
```

Или вот так:

```
denis@MyLinux:~/Lesson16$ chmod a=r denis.txt
```

А как сделать подобные изменения числовым способом?

Стилки бит

Создайте директорию с неограниченными правами доступа для всех пользователей, например MySHARE:

```
denis@MyLinux:/home$ ls -l
total 24
drwxr-xr-x 22 denis denis 4096 Nov 22 21:49 denis
drwxr-xr-x 2 kolya kolya 4096 Nov 20 12:37 kolya
drwxrwxrwx 2 root root 4096 Nov 22 22:19 MySHARE
drwxr-xr-x 3 petya petya 4096 Nov 20 14:07 petya
drwxr-xr-x 4 user2 user2 4096 Nov 20 13:57 user2
drwxr-xr-x 3 vasya vasya 4096 Nov 22 21:50 vasya
denis@MyLinux:/home$ cd MySHARE/
```


Ссылки на файлы

Попробуем поиграть со ссылками. Сначала займемся жесткими.

Создайте несколько жестких ссылок в разных директориях для нашего файла:

```
$ ln fun fun-hard
$ ln fun dir1/fun-hard
$ ln fun dir2/fun-hard
```

Теперь у нас есть четыре экземпляра файла fun. Посмотрим, что содержит наш каталог:

```
$ ls -li
итого 16
12353539 drwxrwxr-x 2 me me 4096 2012-01-14 16:17 dir1
12353540 drwxrwxr-x 2 me me 4096 2012-01-14 16:17 dir2
12353538 -rw-r--r-- 4 me me 1650 2012-01-10 16:33 fun
12353538 -rw-r--r-- 4 me me 1650 2012-01-10 16:33 fun-hard
```

Следует обратить внимание на второе поле в записях, соответствующих файлам fun и fun-hard. Оба они содержат 4 – число жестких ссылок на файл, существующих в данный момент. Как вы помните, файл всегда имеет хотя бы одну жесткую ссылку, потому что имя файла определяется ссылкой. Но как убедиться, что fun и fun-hard – это один и тот же файл? В этом случае команда ls нам не помощник. Вы, конечно, скажете, что fun и fun-hard имеют одинаковые размеры (поле 5), но по списку файлов нельзя уверенно утверждать, что это один и тот же файл. Обратите внимание на *иноды* двух файлов, они одинаковые. Это значит, что имена файлов ссылаются на одну область данных.

Удаление файлов и каталогов выполняется при помощи команды rm. Далее мы немного почистим нашу песочницу. Сначала удалите одну из жестких ссылок:

```
$ rm fun-hard
$ ls -li
итого 12
drwxrwxr-x 2 me me 4096 2012-01-15 15:17 dir1
lrwxrwxrwx 1 me me 4 2012-01-16 14:45 dir1-sym -> dir1
drwxrwxr-x 2 me me 4096 2012-01-15 15:17 dir2
-rw-r--r-- 3 me me 1650 2012-01-10 16:33 fun
lrwxrwxrwx 1 me me 3 2012-01-15 15:15 fun-sym -> fun
```

Результат получился вполне ожидаемым. Файл `fun-hard` исчез, и счетчик ссылок во втором поле в записи для файла `fun` уменьшился с четырех до трех. Далее, удалите файл `fun` и ради развлечения добавьте в команду параметр `-i`, чтобы посмотреть, что происходит:

```
$ rm -i fun
```

`rm: удалить обычный файл `fun'?`

Введите `y` в ответ на запрос, и файл будет удален. Посмотрим на ввод `ls`, что произошло с `fun-sym`? Поскольку теперь Символическая ссылка указывает на несуществующий файл, она стала *битой*:

```
$ ls -l
```

итого 8

`drwxrwxr-x 2 me me 4096 2012-01-15 15:17 dir1`

`lrwxrwxrwx 1 me me 4 2012-01-16 14:45 dir1-sym -> dir1`

`drwxrwxr-x 2 me me 4096 2012-01-15 15:17 dir2`

`lrwxrwxrwx 1 me me 3 2012-01-15 15:15 fun-sym -> fun`

Обратите внимание как в вашем дистрибутиве отображена битая ссылка. Битые ссылки не представляют никакой опасности, но вносят определенную путаницу. При попытке использовать битую ссылку вы увидите:

```
$ less fun-sym
```

`fun-sym: Нет такого файла или каталога`

Попробуем удалить символическую ссылку:

```
$ rm fun-sym dir1-sym
```

```
$ ls -l
```

итого 8

`drwxrwxr-x 2 me me 4096 2012-01-15 15:17 dir1`

`drwxrwxr-x 2 me me 4096 2012-01-15 15:17 dir2`

Главное, что следует помнить о символических ссылках: большинство операций с файлами воздействуют на целевой элемент, а не на саму ссылку. Однако команда `rm` является исключением из этого правила. Когда вы удаляете ссылку, удаляется сама ссылка, а не элемент, на который она указывает.

Идея ссылок на первый взгляд может показаться малопонятной, поэтому уделите время их исследованию. Зачастую они оказываются настоящим спасательным кругом.

Давайте немного приберем за собой, удалим учебный каталог-песочницу *playground*. Для этого вернитесь в домашний каталог и вызовите команду `rm` с параметром рекурсивного удаления каталогов (`-r`), чтобы удалить каталог *playground* и все его содержимое, включая подкаталоги:

```
$ cd
```

```
$ rm -r playground
```

Best of LUCK with it, and remember to HAVE FUN while you're learning :)
Sergey Stankevich



ЗАДАНИЯ

Задание 1 – Создание пользователей

Создайте *стандартного пользователя* с помощью GUI. Пользователю присвойте имя члена вашей команды, но в имени должен присутствовать суффикс **GUI*. Это нужно для проведения экспериментов. При присвоении имен действуйте по принципу: «Относитесь к именам переменных, как к именам детей своих». Пароль пользователя должен быть простым, например «123».

Небольшой совет. Имена пользователей пишите с маленьких букв, это поможет ускорить вашу дальнейшую работу. Разделителем в имени используйте нижнее подчеркивание «_», тире «-» или заглавную букву.

Например, serhey_stankevich, sergei-stankewich, sergStankevich.

Просмотрите содержимое файла **/etc/passwd**, сравните атрибуты *реальных пользователей* и пользователя **root**. Результат подтвердите скриншотом. Дайте пояснения.

При создании пользователей, которые нужны для экспериментов, назначайте им одинаковые пароли (это тоже эксперимент), такие чтобы не забыть. Например, «123».

Создайте несколько *стандартных пользователей* (аккаунтов) посредством *командной строки*. При этом пользователям присвойте имена членов вашей команды, но в имени должен присутствовать суффикс **CL* или **CLI*. Это нужно для того, чтобы отличить пользователей созданных при помощи графического и консольного интерфейсов.

Создайте двух пользователей с одинаковыми простыми паролями.

Создайте *скелет* и *пользователя с шаблоном скелета* в директории **/home**.

Также попробуйте создать аккаунт реального пользователя без пароля.

Задание 2 – Изменение параметров паролей пользователей

Пароли должны быть простые и **одинаковые**. Это нужно для проведения экспериментов. Просмотрите содержимое файла **/etc/passwd**, сравните атрибуты *реальных пользователей*. Просмотрите содержимое файла **/etc/shadow**, сравните атрибуты паролей пользователей, особое внимание обратите на шифр пароля у пользователей с одинаковым паролем.

Измените периоды изменения паролей для пользователей. Внесите ограничения в следующие атрибуты: минимальный возраст пароля (сутки); максимальный возраст пароля (сутки); период предупреждения пароля; период бездействия пароля; дата истечения срока действия аккаунта.

Результат подтвердите скриншотом. Дайте пояснения.

Задание 3 – Создание групп и работа с правами доступа к файлам

Создайте пользователя (аккаунт) с правами администратора. В каталогах **/home/** администраторов и других пользователей создайте по несколько файлов с разными правами доступа.

Создайте общую группу пользователей.

Выполните следующие требования:

1. Одного из пользователей перевести в группу **shadow**
2. Создать у каждого пользователей директорию с 2 файлами
3. Просмотреть текущие права доступа к файлам для всех пользователей
4. Каталог пользователя в группе **shadow** сделать доступным только в своей группе
5. Файлы второго пользователя сделать доступными только владельцам
6. Под админом назначить всем созданным файлам права только для чтения для всех пользователей
7. Пользователем в группе **shadow** лишить всех остальных пользователей права исполнять его файлы
8. Под админом назначить всем пользователям все права
9. Удалить пользователя, находящегося в группе **shadow**

Укажите является ли группа системной или создана пользователем.

Получите идентификаторы пользователей и состояние активных пользователей системы. Проверьте содержимое файлов **/etc/shadow**, **/etc/passwd**, **/etc/group**.

Некоторые файлы защитите *липким битом*. Попробуйте удалить созданные файлы из других аккаунтов администраторов и простых пользователей. Опишите какой получили результат.

Создайте *жесткую ссылку*, перенесите эту ссылку в пространство другого пользователя и откройте ее в сеансе этого пользователя, затем присвойте жесткой ссылке одну группу и откройте ссылку из пользователя этой группы. Измените владельца и группу жесткой ссылки и посмотрите, как изменились атрибуты основного файла.

Примените разные варианты изменения атрибутов доступа файла и каталогов. Попробуйте совершить разные операции с этими файлами от имени других пользователей. Активно используйте команду **su**.

Не бойтесь выйти за рамки предложенных упражнений и заданий – добавьте дополнительные файлы и каталоги, поэкспериментируйте с групповыми символами для определения групп файлов в разных операциях.

«Easy things should be easy and hard things should be possible»
«Простые вещи должны быть простыми, а сложные вещи должны быть возможными»



Контрольные вопросы:

- 1) Какая концепция прав доступа к файлу реализована в ядре?
- 2) Какие типы пользователей существуют в Linux (UNIX)?
- 3) Поясните понятие «реальный пользователь».
- 4) В чем различие понятий «пользователь» и «владелец файла»?
- 5) Что такое «аккаунт» или «логин», из каких атрибутов состоит?
- 6) Опишите отличия команд **su** и **sudo**.
- 7) В чем отличие символов строки приглашения **\$** и **#**?
- 8) Для чего предназначена системная директория **etc**, какие файлы в ней хранятся?
- 9) Какую информацию содержат конфигурационные файлы **passwd** и **shadow**?
- 10) В чем различие понятий «пользователь» и «аккаунт пользователя»? Объясните это на примере строки файла **/etc/passwd**.
- 11) Поясните атрибуты пользователя записанные в файле **/etc/passwd**.
- 12) Поясните атрибуты пароля пользователя в файле **/etc/shadow**.
- 13) Что такое липкий бит (Sticky bit), для чего и как он применяется?
- 14) Что такое иоды файлов?
- 15) Что такое жесткая ссылка и какие ограничения она имеет?
- 16) Что такое символическая ссылка, что такое «битая» ссылка?
- 17) Каком образом в вашем дистрибутиве отображена битая ссылка?

Дополнительная и справочная информация

Шоттс У. Ш80 Командная строка Linux. Полное руководство. — СПб.: Питер, 2017. — 480 с.: ил. — (Серия «Для профессионалов»).

102 Глава 9. Привилегии

38 Глава 3. Исследование системы – Определение типов файлов командой **file**

Таблица 9.1. Типы файлов

Атрибут	Тип файла
-	Обычный файл
d	Каталог
l	Символическая ссылка. Обратите внимание, что для символических ссылок все остальные атрибуты имеют значение gwxgwxgwx и не отражают действительные права доступа. Фактические права доступа к файлу определяются атрибутами самого файла, на который указывает символическая ссылка
c	Специальный файл символьного устройства. Файлы этого типа соответствуют устройствам, таким как терминал или модем, которые обрабатывают данные как потоки байтов
b	Специальный файл блочного устройства. Файлы этого типа соответствуют устройствам, таким как привод жесткого диска или CD-ROM, которые обрабатывают данные блоками

Таблица 9.2. Атрибуты прав доступа

Атрибут	Файлы	Каталоги
r	Разрешается открывать и читать содержимое файла	Разрешается читать содержимое каталога, если вместе с этим атрибутом установлен атрибут права на выполнение
w	Разрешается записывать в файл или усекаать его; однако этот атрибут не дает права переименовывать и удалять файлы. Возможность переименования и удаления файлов определяется атрибутами вмещающего каталога	Разрешается создавать, удалять и переименовывать файлы внутри каталога, если вместе с этим атрибутом установлен атрибут права на выполнение
x	Разрешается интерпретировать файл как программу и выполнять ее. Файлы, содержащие программы на языках сценариев, дополнительно должны быть доступны для чтения, иначе они не будут выполняться	Разрешается входить в каталог, то есть выполнять команду <code>cd</code> для перехода в него

Таблица 9.3. Примеры установки атрибутов прав доступа к файлам

Атрибуты файлов	Значение
-rwx-----	Обычный файл, доступный владельцу для чтения, записи и выполнения. Никто другой не имеет прав доступа к файлу
-rw-----	Обычный файл, доступный владельцу для чтения и записи. Никто другой не имеет прав доступа к файлу
-rw-r--r--	Обычный файл, доступный владельцу для чтения и записи. Члены группы имеют право читать файл. Все остальные имеют право читать файл
-rwxr-xr-x	Обычный файл, доступный владельцу для чтения, записи и выполнения. Все остальные имеют право читать и выполнять файл
-rw-rw----	Обычный файл, доступный для чтения и записи только владельцу и членам группы
Lrwxrwxrwx	Символическая ссылка. Все символические ссылки имеют недействительные значения атрибутов. Фактические права доступа к файлу определяются атрибутами самого файла, на который указывает символическая ссылка
drwxrwx---	Каталог. Владелец и члены группы могут входить в каталог, создавать, переименовывать и удалять файлы внутри каталога
drwxr-x---	Каталог. Владелец может входить в каталог, создавать, переименовывать и удалять файлы внутри каталога. Члены группы могут входить в каталог, но не могут создавать, переименовывать и удалять файлы внутри каталога

Таблица 9.5. Символическая форма записи аргументов команды `chmod`

Символ	Значение
u	Сокращенно от <i>user</i> (пользователь), означает владельца файла или каталога
g	Группа
o	Сокращенно от <i>other</i> (другие, остальные), означает весь остальной мир
a	Сокращенно от <i>all</i> (все); комбинация из всех трех символов: <i>u</i> , <i>g</i> и <i>o</i>

Таблица 9.6. Примеры символической формы записи прав доступа к файлам

Атрибуты файлов	Значение
u+x	Добавляет право на выполнение, но только для владельца
u-x	Отнимает право на выполнение у владельца
+x	Добавляет право на выполнение для владельца, группы и остального мира. Эквивалент записи <code>a+x</code>
o-rw	Отнимает право на чтение и запись у всех, кроме владельца и группы
go=rw	Устанавливает право на чтение и запись для всех, кроме владельца. Если прежде файл имел разрешение на выполнение для группы и всего мира, это право отнимается
u+x, go=rwx	Добавляет право на выполнение для владельца и устанавливает право на чтение и выполнение для группы и всего мира. При выполнении сразу нескольких операций с привилегиями они должны разделяться запятой

Иванов Н. Н. И20 Программирование в Linux. Самоучитель. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2012. — 400 с.: ил.

78 Глава 7. Базовые операции ввода-вывода
105 chmod — изменение режима доступа к файлу

Робачевский А. М. Операционная система UNIX®. - СПб.: 2002. - 528 ил.

50 Глава 7. Базовые операции ввода-вывода

Подробно песочница представлена к книге: Шоттс У. «Командная строка Linux. Полное руководство.» — СПб.: Питер, 2017. — 480 с.: ил. — (Серия «Для профессионалов»). на страницах 225-226.

<https://ru.stackoverflow.com/questions/800756/Группа-администраторов-в-ubuntu/800776>

Группа adm отвечает за доступ к некоторым логам в /var/log:

```
# посмотрим на типичный лог файл
ls -la /var/log/dmesg
# ответ:
-rw-r----- 1 root adm 58692 марта 17 12:24 /var/log/dmesg
```

посмотрим группы syslogа:

```
groups syslog
# ответ:
syslog : syslog adm
```

Группа sudo разрешает запускать с правами суперпользователя - это механизм, чтобы [не работать под рутом](#). sudo можно настроить:

```
sudo cat /etc/sudoers
# User privilege specification
root  ALL=(ALL:ALL) ALL
```

```
# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
```

```
# Allow members of group sudo to execute any command
%sudo  ALL=(ALL:ALL) ALL
```

Таким образом, в `sudo` прописано, что `root`, и пользователи групп `admin` и `sudo` имеют суперпривилегии. Группа `admin` появилась исторически, и оставлена для обратной совместимости. У меня на 14 убунте группы нет, но в `sudoers` она упоминается.

Для редактирования этого файла нужно ввести `sudo visudo`

Какая группа отвечает за группу администраторов?

Нет понятия "система отличает админа от обычного юзера". Система более гибкая, все правила изложены в файле `sudoers`.

Суперадмин `root` по-умолчанию в Убунте отключен. При установке создается пользователь, который включен в группу `sudo`, и через `sudo {команда}` может выполнять любые административные задачи.

Но затем `sudo` можно настроить так, что любой пользователь сможет выполнять любые, даже административные действия. И даже снести систему. И для этого не нужно включать пользователя в какую-то группу, достаточно "как надо" настроить файл `sudoers`.

Пример. Мне нужно перезагружать Sphinx из веб-интерфейса, но я не хочу давать абсолютно все права веб-серверу (это не безопасно). Через `sudoers` я могу это сделать.

Считайте, что группа администраторов - это группа `sudo`. Однако в Linux, путем модифицирования `/etc/sudoers` можно сделать любого пользователя администратором, либо любую группу администраторами.

По сути, не важно какая группа является админской, важно только - какая группа (или пользователь) имеет возможность работать с `root` правами.

<https://poweruser.guru/questions/456762/список-администраторов-в-linux>

Чтобы узнать, кто обычно имеет права `sudo`, вы можете просмотреть файл `sudoers`, который находится в `/etc/sudoers`. Вы также можете посмотреть формат файла на страницах руководства. По умолчанию в Ubuntu есть группа `admin`, и любой в этой группе может использовать `sudo`. В других дистрибутивах может быть группа `sudoers` группа `sudo` или группа `staff`.

Помимо этого, существует множество «частично-административных» вещей, которые можно настроить. `Root` является только "администратором", потому что он по своей сути обходит проверки безопасности. Если у пользователя есть определенные разрешения для определенной функции, то он также является "администратором" в этой области; Например, точно так же, как вы являетесь администратором в своем домашнем

каталоге, вы можете назначить кого-то администратором через веб-сервер или администратором через видео / аудио выходы на компьютере (посмотрите `video` и `audio` группы, если я правильно помню Ubuntu).

Подводя итог, не совсем простой способ получить "список администраторов". Однако, чтобы быстро получить большинство из них:

- `root` , просто потому что.
- Любой с привилегиями `sudo` или в группе с привилегиями `sudo`. Обычно кто-либо из групп `admin` , `wheel` , `staff` , `sudo` или `sudoers` , если они существуют.
- Конечно, любой, кто знает пароль к вышеупомянутым аккаунтам

В Linux специальные привилегии обычно предоставляются через группы. Например, люди из группы `sudoers` могут использовать `sudo`, люди из группы `audio` могут воспроизводить аудио и т.д. Используйте команду `groups` чтобы просмотреть группы, в которых находится пользователь.

Для того, чтобы на самом деле найти администратор для группы, вы бы взглянуть на файл - `group` - В этом файле перечислены все администраторы для каждой группы в списке. Вы можете только просматривать содержимое этого файла , если вы админ повышен через `sudo` или `root` Обычный файл группы, как упоминали другие, будет перечислять только общее членство в таких перечисленных группах.

Интернет-источники

https://help.ubuntu.ru/wiki/руководство_по_ubuntu_server/безопасность/user_management

<https://techlist.top/linux-users-types-of-users/>

<https://man7.org/linux/man-pages/>

Создание пользователя в Linux

[Как создать пользователя Linux - Losst](#)

[Создание пользователя в Linux. Команды `adduser` и `useradd`. Linux статьи \(pingvinus.ru\)](#)

4 способа как установить права администратора в Linux:

[Как дать Root права пользователю в Linux | Пошаговая Инструкция \(setiwick.ru\)](#)

Удаление пользователя

[Как удалить пользователя в Linux - Losst](#)

Команда `su`

["Linux по-русски". А.Дмитриев. "Что мы знаем о команде su?" \(rus-linux.net\)](#)

`/etc/passwd` - тут хранятся все аккаунты
`/etc/shadow` - тут хранятся все пароли аккаунтов
`/etc/group` - тут хранятся все группы



We hope you enjoy working with Linux!

