

# Linux: о пользователях

При установке операционной системы вы создавали пользователя и видели, что от этого же пользователя запущен Telegram. Кроме того, вы могли заметить, что, когда вы смотрели список процессов, некоторые из них были запущены из-под пользователя **root**.

Ещё вы уже пользовались командой **sudo**, когда устанавливали окружение рабочего стола и Telegram.

**Пользователь** — это некто или нечто, которые обладают именем, идентификатором и набором прав (например, имеют доступ к каким-то файлам или могут запускать какие-то программы).

Имя (для удобства человека) и PID (численный идентификатор для удобства компьютера) имеют одну и ту же функцию: и то и другое — уникальное имя, позволяющее отличать одного пользователя от другого.

Когда вы что-то делаете в операционной системе, вы делаете это от имени своего пользователя.

Пользователем может быть человек, а может быть программа. То есть это не обязательно идентификатор конкретной личности, как в случае с логином для Telegram. Это скорее абстрактная сущность, описывающая объект и те права, которые у него есть. Пользователи объединяются в группы.

Если вы вдруг забыли, как зовут пользователя, под которым вы залогинились, Linux может вам об этом напомнить:

```
whoami
```

В Linux есть особенный пользователь с неограниченными правами — так называемый **суперпользователь**. Он называется **root**. У него неограниченные права, неограниченные возможности что-то сделать в системе.

С правами пользователя root можно испортить приложения, которые работают на сервере, удалить важные данные, нанести вред самой операционной системе. Поэтому логиниться пользователем root хоть технически возможность и есть (но, кстати, в вашем дистрибутиве, ubuntu, она по умолчанию отключена), но на практике это считается дурным тоном.

Обычный пользователь не может устанавливать приложения. В процессе установки модифицируются и появляются системные файлы. Для этого нужны права суперпользователя.

Для того чтобы устанавливать приложения под логином обычного пользователя, нужна команда **sudo**. Она позволяет выполнять действия с привилегиями суперпользователя (на самом деле, с привилегиями и других пользователей тоже). Например, вы можете воспользоваться этим, если вам нужно выполнить какие-то действия от имени одной из программ.

Основным объектом прав в Linux являются файлы.

Давайте вспомним, что вы видели в выводе команды **ls**, сейчас нам станет понятнее, что значат первые колонки.

```
-rw-r--r--  1 vmarkova vmarkova 220 Apr  9 2014 .bash_logout
-rw-r--r--  1 vmarkova vmarkova 3637 Apr  9 2014 .bashrc
drwx---  2 vmarkova vmarkova 4096 Mar  4 16:41 .cache
```

Третья и четвёртая колонки — это пользователь и группа, которые являются владельцами файла (или директории). Первая колонка — это собственно права, которые присвоены файлу (или директории).

Первая колонка — самая интересная. Структура записи в ней такова: для владельца пользователя, владельца группы и остальных пользователей может выставляться комбинация из прав, **rwX**:

- **read** — просмотр содержимого файла;
- **write** — изменение файла;
- **execute** — запуск файла.

## Каталоги

Чтение позволяет получить имена (только имена, **ls**) файлов в каталоге. А вот чтобы иметь возможность выполнять команду **ls -la**, потребуется право на выполнение. Это связано с тем, что нам потребуется служебная информация. Для того чтобы перейти в каталог, также потребуется право на выполнение.

Для записи в каталог, то есть чтобы иметь возможность создавать файлы, вам потребуются права и на запись, и на выполнение. Потому что мы должны иметь возможность не только что-то добавить, но и получить доступ к служебной информации. Права на запись и на исполнение в случае каталогов почти всегда идут вместе. Либо выставляются оба, либо не выставляется ни один.

Для изменения прав используется команда **chmod**.

Давайте попробуем пару сценариев.

Уберём права выполнения для файла (a-x).

Уберём все права для группы g-rwx.

Права для разных типов пользователей можно перечислять через запятую.

```
Chmod u=rw, g=r, o=r myfirstfile
```

Создадим тёмный каталог.

```
mkdir darkroom  
chmod a-r+x darkroom  
touch darkroom/darkfile  
ls -l darkroom  
ls darkfile
```

Для выполнения практического задания будет полезна ещё одна команда — **rm** (что значит remove — «удалить»):

```
rm myfirstfile  
rm -r myfirstdir
```