Программы в Linux

Права в Linux. Установка программ. Создание и запуск скриптов Python в Linux.

**Оглавление**

[Права в Linux](#_m4qgx6i25ys8)

[Информация о пользователе](#_iv0yq159ssia)

[Команда sudo](#_8d6d7vo2txk4)

[Создание пользователя](#_7upscwaykmk7)

[Команда chmod](#_fq14xowcr3nj)

[Установка программ](#_ws259kworlak)

[Создание и запуск скриптов Python в Linux](#_q3lpm4gqcvdw)

[Практическое задание](#_111kx3o)

[Дополнительные материалы](#_3l18frh)

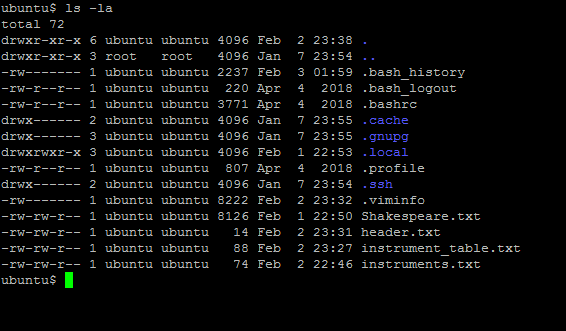
[Используемая литература](#_206ipza)

# 

# 

# Права в Linux

В Linux пользователь обычно принадлежит к группе. Права на совершение действий можно предоставлять как отдельным пользователям, так и группам. Посмотрим на информацию о файлах в домашней директории, запустив команду **ls —la** (добавление параметра **a** выведет также и скрытые файлы):



В каждой строке отражена информация о файле или папке. Если строка начинается с символа **–**, то это обычный файл, а если с символа **d**, то это папка.

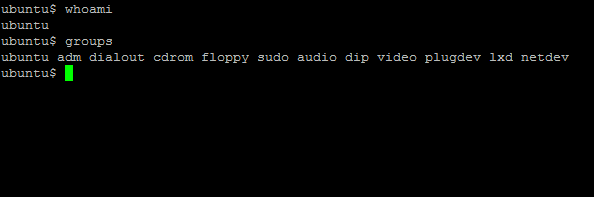
Следующие 9 символов свидетельствуют о наличии прав на этот файл. Первые три из этой девятки говорят о правах владельца на файл, следующие три — о правах его группы, и последние три — о правах остальных пользователей. В каждой тройке первый символ — это **r** (есть права на чтение) или **–** (нет прав на чтение), второй — **w** (есть права на запись) или **–** (нет прав на запись), третий — **x** (есть права на запуск исполняемого файла) или **–** (нет прав на запуск).

В строке с информацией о файле также указан его владелец и группа, к которой принадлежит последний.

Например, для файла **header.txt** указана следующая информация: читать и редактировать файл могут владелец и его группа, остальные пользователи могут только читать. Владельцем файла **header.txt** является пользователь **ubuntu**, состоящий в группе **ubuntu**.

## Информация о пользователе

Посмотрим информацию о текущем пользователе с помощью команды **whoami** и группах, в которых он состоит, посредством команды **groups**:

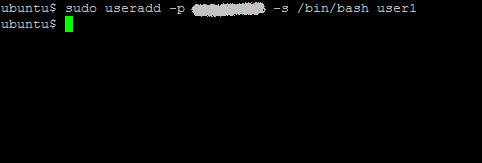


## Команда sudo

Для запуска команд, требующих прав пользователя **root** (это пользователь с высшими правами), вначале нужно указать команду **sudo**. Употребление этой команды означает, что текущий пользователь на время приобретает наибольшие права. Поэтому применять **sudo** нужно аккуратно и избегать непроверенных команд, способных нанести вред системе, хранящимся на сервере данным или нарушить безопасность, открыв доступ к серверу посторонним лицам. Также нужно знать, что права на запуск команд под пользователем есть не у всех.

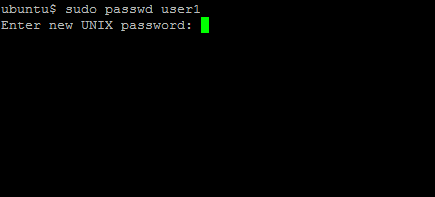
## Создание пользователя

Рассмотрим использование **sudo** на примере команды **useradd**, с помощью которой можно создать нового пользователя. Для этого запустим команду **sudo useradd —p \*\*\*\*\*\*\* -s /bin/bash user1** (звездочками отмечен пароль, который будет у нового пользователя):

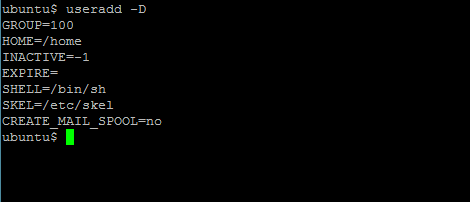


С помощью данной команды мы создали пользователя с именем **user1** и задали пароль для него (указан после опции **–p**), а также оболочку **/bin/bash** для пользователя (после опции **–s**). Также можно задавать основную группу (опция **–g**) и дополнительные группы (опция **–G**), в которых будет состоять новый пользователь. Но если эти опции не задать, то они будут применены со значениями по умолчанию.

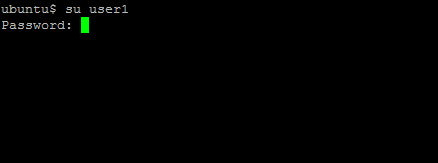
Поменять пароль пользователя **user1** можно с помощью команды **sudo passwd user1**:



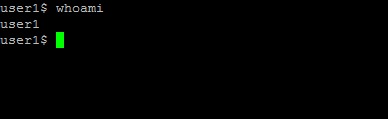
Посмотреть, какие параметры будут даны пользователю после создания по умолчанию, можно с помощью команды **useradd –D**:



Теперь посмотрим, как можно совершать действия под другим пользователем. Для этого нужно перед именем другого пользователя указать команду **su**. Запустим команду **su user1**, чтобы сменить текущего пользователя на **user1**:



Появилось предложение ввести пароль. Вводим пароль нового пользователя и после этого смотрим с помощью команды **whoami**, какой пользователь является текущим:

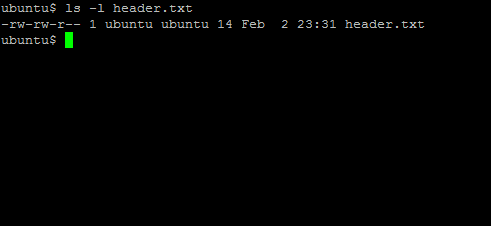


Видим, что текущим пользователем является **user1**, и теперь можем работать под ним. Чтобы вернуться к пользователю, под которым мы зашли на сервер (**ubuntu**), можно ввести команду **exit** и еще раз запустить команду **whoami**, чтобы удостовериться, что текущим пользователем снова стал **ubuntu**.

## Команда chmod

Научимся работать с командой **chmod**, дающей возможность предоставлять права доступа на определенные файлы.

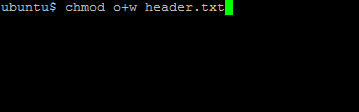
Посмотрим, какие права выставлены для файла **header.txt**, применив команду **ls -l header.txt**:



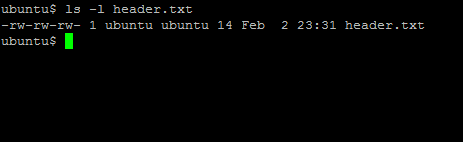
Видим, что для владельца файла и его группы предоставлены права чтения и записи в **header.txt**, а для остальных пользователей есть только права на чтение.

В команде **chmod** предусмотрены категории для обозначения тех, кому мы будем предоставлять или у кого мы будем забирать права. Категория **u** — это владелец файла, **g** — группа файла, **o** — остальные пользователи. После категории указывается знак **+**, если права предоставляются, или **–**, когда права забираются. После этого указываются права, с которыми производится действие — **r**, **w** или **x** (чтение, запись, исполнение).

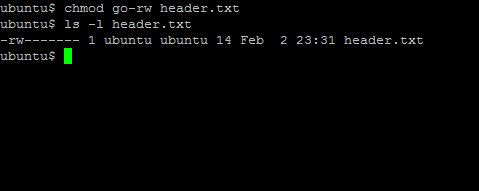
Например, можно предоставить другим пользователям права на запись в файл **header.txt** командой **chmod o+w header.txt**:



После этого еще раз запустим команду **ls -l header.txt** и убедимся, что права на файл поменялись:



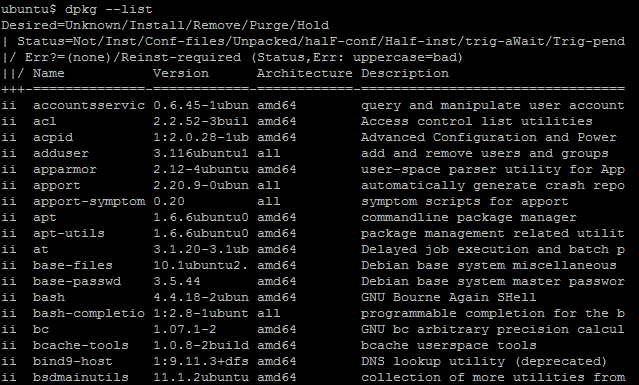
А теперь посмотрим, как можно менять несколько прав для ряда категорий. Например, заберем права на чтение и запись в файл **header.txt** у группы файла и остальных пользователей с помощью команды **chmod go-rw header.txt** и затем проверим, поменялись ли права:



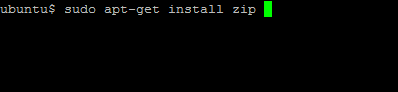
Видим, что права на чтение и запись остались только у владельца файла.

# Установка программ

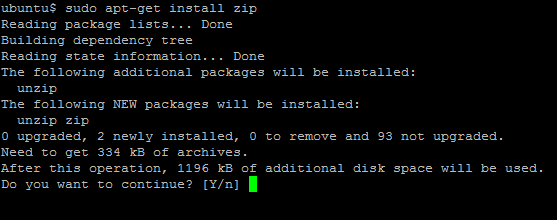
Для просмотра списка всех программ, установленных на данном сервере, нужно запустить команду **dpkg -l** или **dpkg --list**:



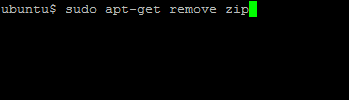
Допустим, у нас нет программы **zip**. Установить ее можно с помощью команды **sudo apt-get install zip**:



После запуска команды может потребоваться подтверждение. Нужно будет ввести **Y** и нажать **Enter**:



Для удаления программы **zip** нужно будет ввести команду **sudo apt-get remove zip**:

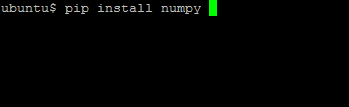


Затем для подтверждения потребуется ввести **Y** и нажать **Enter**. После этого запустится удаление программы.

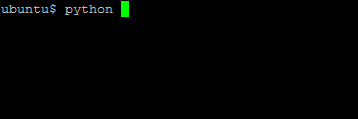
# Создание и запуск скриптов Python в Linux

Python уже должен быть установлен в Ubuntu, так как входит в стандартный дистрибутив этой ОС. Чтобы создать программу, написанную на Python, потребуется библиотека **Numpy** (название произносится как «нампа́й») — она не встроенная, поэтому ее нужно будет отдельно установить. Воспользуемся для этого установщиком **pip** (если он не установлен, то это нужно будет предварительно сделать: **sudo apt install python-pip**).

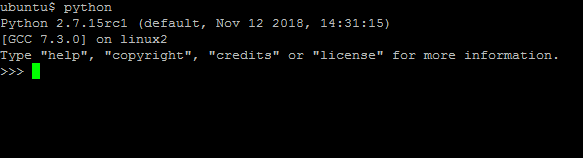
Запустим команду **pip install numpy**:



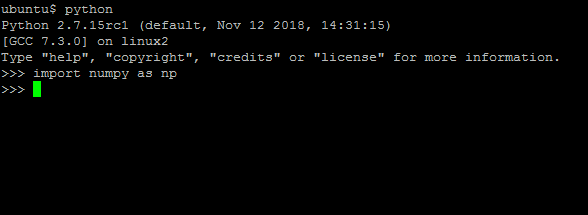
Установив библиотеку **numpy**, запустим **python**:



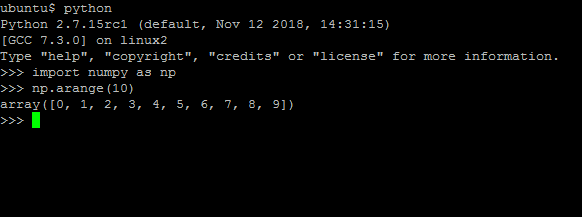
Появится командная строка интерпретатора:



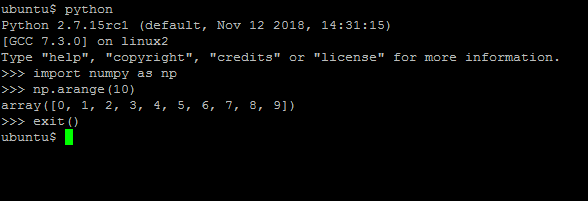
Импортируем библиотеку Numpy с помощью команды **import numpy as np**:



Запустим команду, создающую массив из 10 чисел — **np.arange(10)**:



Выйти из командной строки интерпретатора Python можно с помощью команды **exit()**:



А теперь с помощью текстового редактора **vim** создадим файл **create\_matrix.py**.

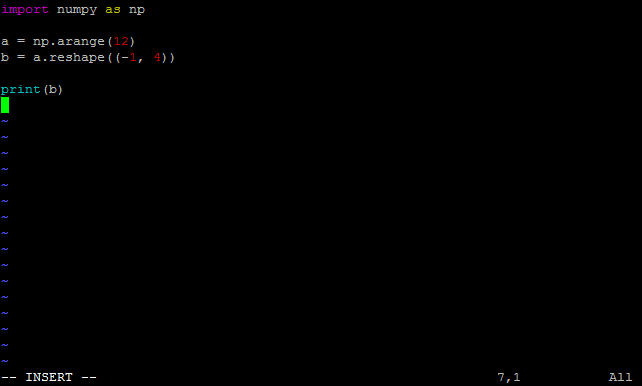
Внесем в него следующую программу:

**import numpy as np**

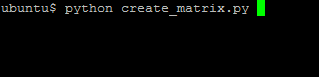
**a = np.arange(12)**

**b = a.reshape((-1, 4))**

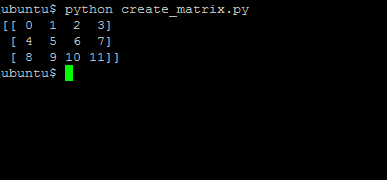
**print(b)**

****

Запустим этот файл, используя команду **python create\_matrix.py**:



После этого на экране должна появиться матрица размером 3 на 4 с числами от 0 до 11:



# Практическое задание

1. Создать пользователя **user\_new** и предоставить ему права на редактирование файла с программой, выводящей на экран ***Hello, world!***
2. Зайти под юзером **user\_new** и с помощью редактора **Vim** поменять фразу в скрипте из пункта 1 на любую другую.
3. \* Под юзером **user\_new** зайти в его домашнюю директорию и создать программу на Python, выводящую в консоль цифры от 1 до 10 включительно с интервалом в 1 секунду.

# Дополнительные материалы

1. [Пишем первую программу на Python в Linux](http://ithowto.ru/164-pishem-pervuyu-programmu-na-python-v-linux.html).
2. [HOW-TO: программа на Python](https://help.ubuntu.ru/fullcircle/27/python_%D1%87_1).
3. [chmod в Linux](https://omgubuntu.ru/chmod/).
4. [Как создать пользователя в Linux](https://losst.ru/kak-sozdat-polzovatelya-linux).

# Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. [Команда chmod Linux](https://losst.ru/komanda-chmod-linux).
2. [Как создать пользователя Linux с помощью командной строки](https://linuxinsider.ru/kak-sozdat-polzovatelya-linux/).
3. [Создание, редактирование и удаление пользователей в Linux](https://www.dmosk.ru/miniinstruktions.php?mini=linux-users).
4. [Как устанавливать программы для Linux](https://lifehacker.ru/kak-ustanavlivat-programmy-dlya-linux/).