GeekBrains

Linux и облачные вычисления

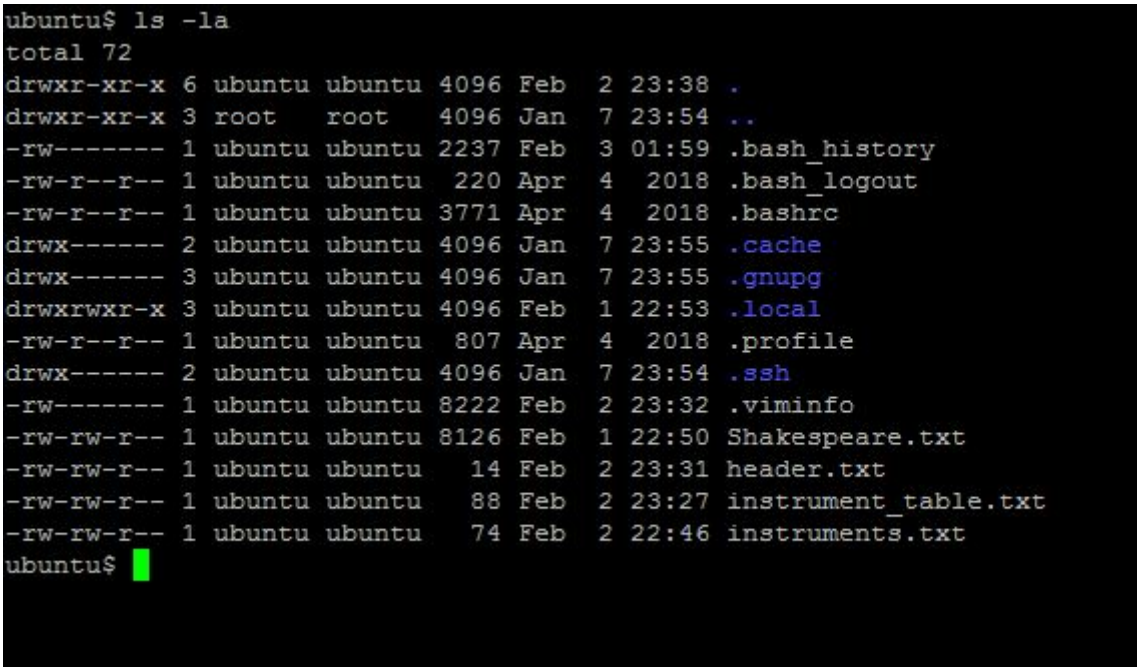
Программы в Linux

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

Оглавление

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.

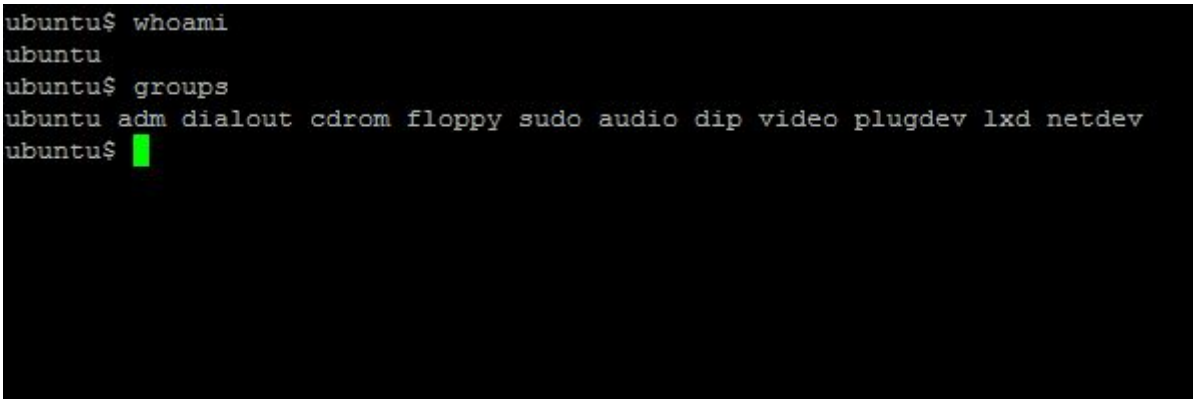
Следующие 9 символов свидетельствуют о наличии прав на этот файл. Первые три из этой девятки говорят о правах владельца на файл, следующие три — о правах его группы, и последние три — о правах остальных пользователей. В каждой тройке первый символ — это **r** (есть права на чтение) или **–** (нет прав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

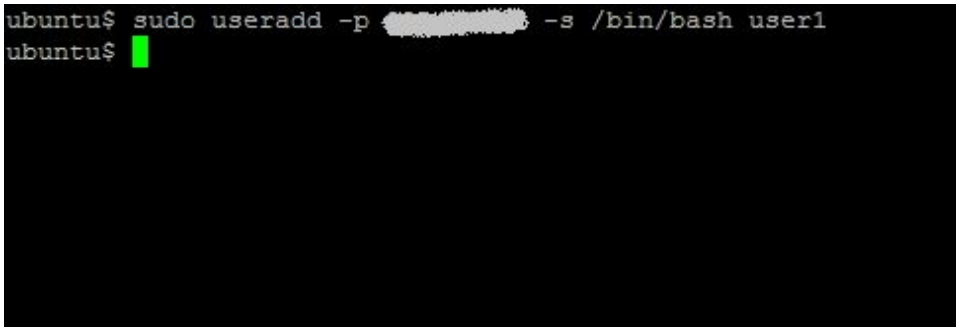


Команда sudo

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

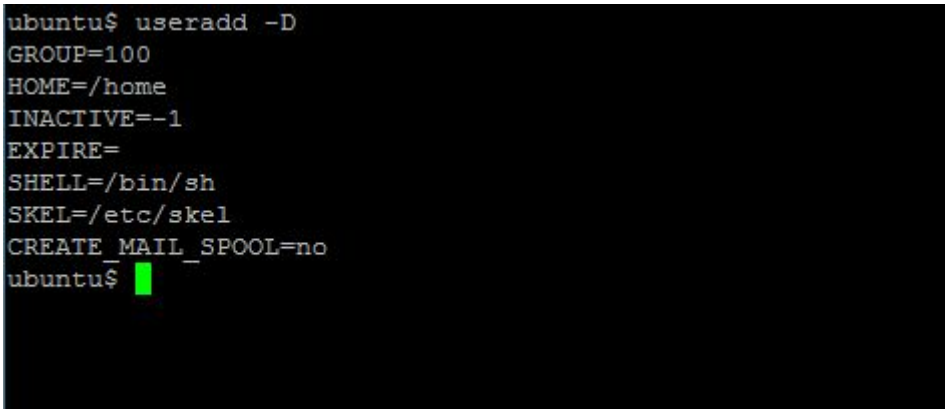
Рассмотрим использование **sudo** на примере команды **useradd**, с помощью которой можно создать нового пользователя. Для этого запустим команду **sudo useradd —p ******* -s /bin/bash user1** (звездочками отмечен пароль, который будет у нового пользователя):



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**:

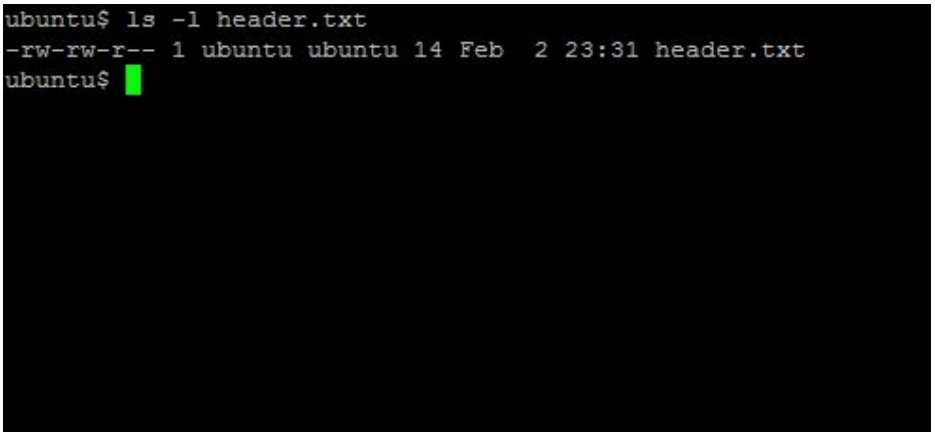
Появилось предложение ввести пароль. Вводим пароль нового пользователя и после этого смотрим с помощью команды **whoami**, какой пользователь является текущим:

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.

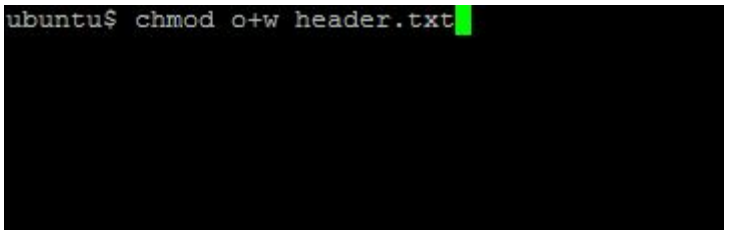
Посмотрим, какие права выставлены для файла **header.txt**, применив команду **ls -l header.txt**:



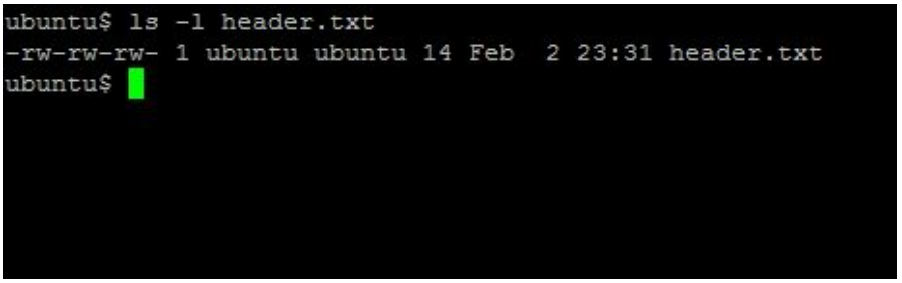
Видим, что для владельца файла и его группы предоставлены права чтения и записи в **header.txt**, а для остальных пользователей есть только права на чтение.

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).

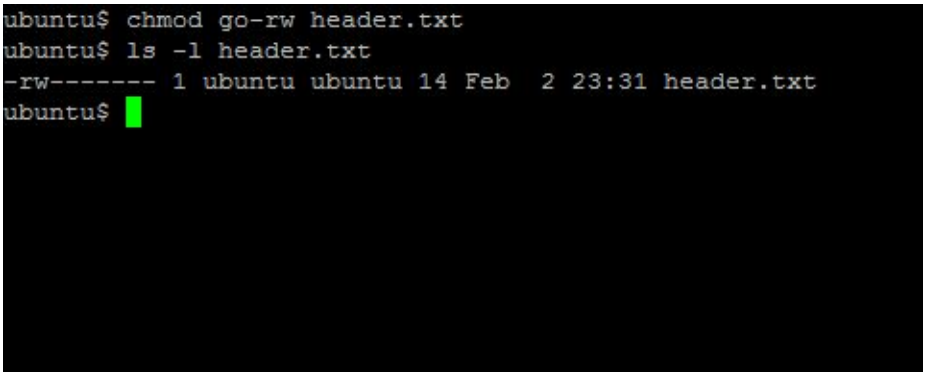
Например, можно предоставить другим пользователям права на запись в файл **header.txt** командой **chmod o+w header.txt**:



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**:

```
ubuntu$ dpkg --list
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name           Version      Architecture Description
+++-==============-============-============-=================================
ii  accountsservic 0.6.45-1ubun amd64        query and manipulate user account
ii  acl            2.2.52-3buil amd64        Access control list utilities
ii  acpid          1:2.0.28-1ub amd64        Advanced Configuration and Power
ii  adduser        3.116ubuntu1 all          add and remove users and groups
ii  apparmor       2.12-4ubuntu amd64        user-space parser utility for App
ii  apport         2.20.9-0ubun all          automatically generate crash repo
ii  apport-symptom 0.20         all          symptom scripts for apport
ii  apt            1.6.6ubuntu0 amd64        commandline package manager
ii  apt-utils      1.6.6ubuntu0 amd64        package management related utilit
ii  at             3.1.20-3.1ub amd64        Delayed job execution and batch p
ii  base-files     10.1ubuntu2. amd64        Debian base system miscellaneous
ii  base-passwd    3.5.44       amd64        Debian base system master passwor
ii  bash           4.4.18-2ubun amd64        GNU Bourne Again SHell
ii  bash-completio 1:2.8-1ubunt all          programmable completion for the b
ii  bc             1.07.1-2     amd64        GNU bc arbitrary precision calcul
ii  bcache-tools   1.0.8-2build amd64        bcache userspace tools
ii  bind9-host     1:9.11.3+dfs amd64        DNS lookup utility (deprecated)
ii  bsdmainutils   11.1.2ubuntu amd64        collection of more utilities from
```

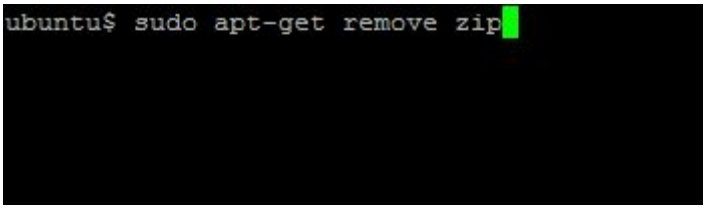
Допустим, у нас нет программы **zip**. Установить ее можно с помощью команды **sudo apt-get install zip**:

```
ubuntu$ sudo apt-get install zip
```

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

```
ubuntu$ sudo apt-get install zip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  unzip
The following NEW packages will be installed:
  unzip zip
0 upgraded, 2 newly installed, 0 to remove and 93 not upgraded.
Need to get 334 kB of archives.
After this operation, 1196 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

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

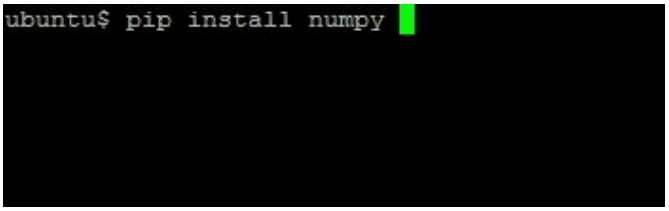


Затем для подтверждения потребуется ввести **Y** и нажать **Enter**. После этого запустится удаление программы.

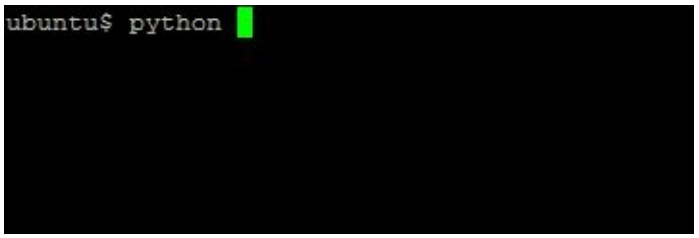
Создание и запуск скриптов Python в Linux

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**).

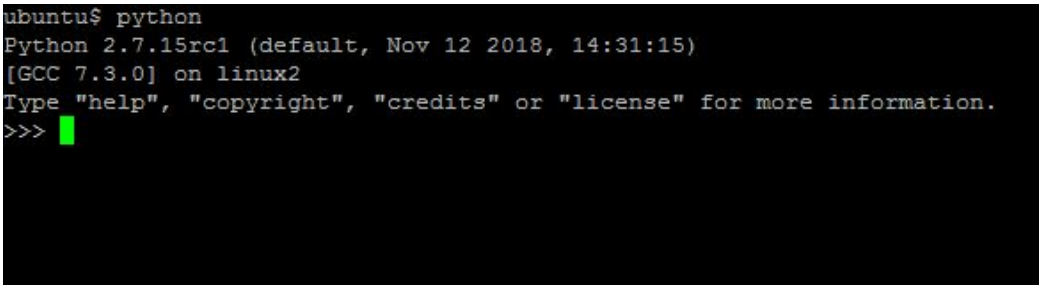
Запустим команду **pip install numpy**:



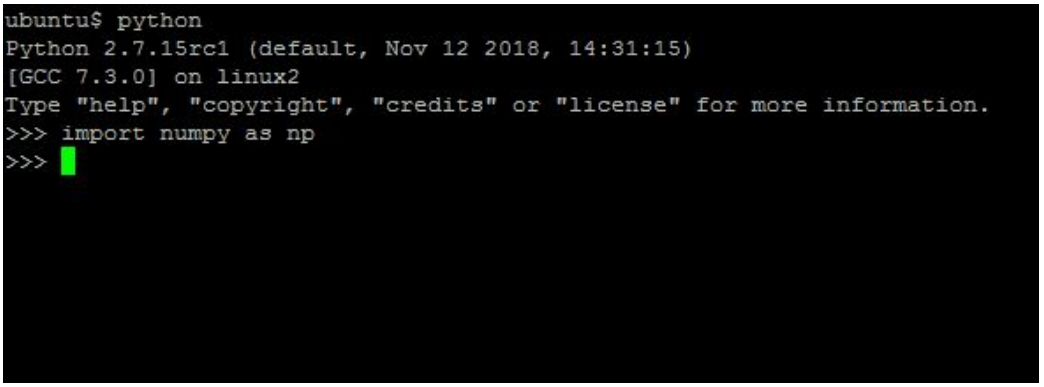
Установив библиотеку **numpy**, запустим **python**:



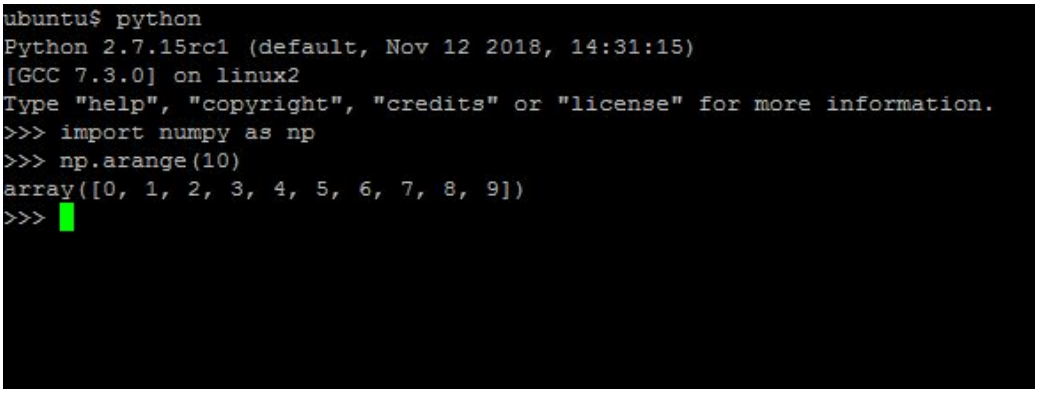
Появится командная строка интерпретатора:



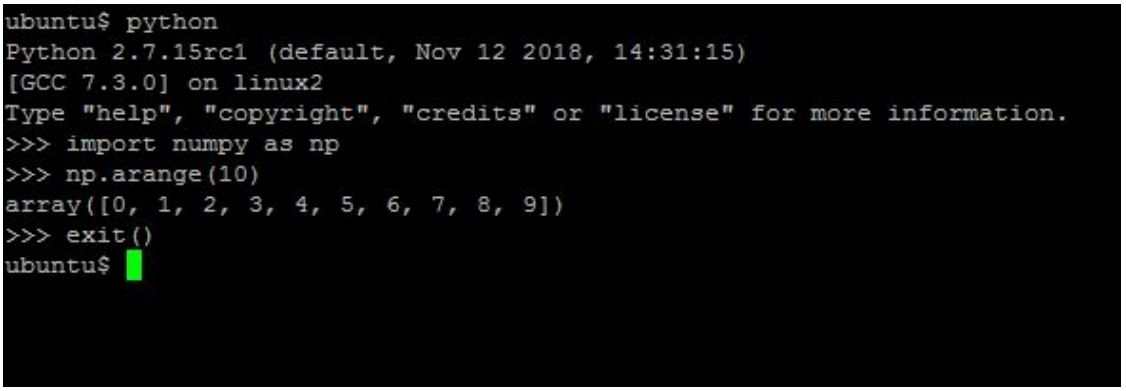
Импортируем библиотеку Numpy с помощью команды **import numpy as np**:



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)
```

```
import numpy as np

a = np.arange(12)
b = a.reshape((-1, 4))

print(b)
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                    7,1           All
```

Запустим этот файл, используя команду **python create_matrix.py**:

```
ubuntu$ python create_matrix.py
```

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

```
ubuntu$ python create_matrix.py
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
ubuntu$
```

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.
2. HOW-TO: программа на Python.
3. chmod в Linux.
4. Как создать пользователя в Linux.

Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. Команда chmod Linux.
2. Как создать пользователя Linux с помощью командной строки.
3. Создание, редактирование и удаление пользователей в Linux.
4. Как устанавливать программы для Linux.