Справочник по командам Linux

В данном мануале собран «теоретический минимум» по командам ОС Linux, необходимым для сетевого администрирования.

Содержание

Все примеры команд в этом справочнике приведены на примере Linux Ubuntu Server 20.04.

Системные требования Ubuntu Server 20.04:
1 GHz CPU
512 MB RAM (system memory)
2.5 GB hard drive

Просмотреть версию Ubuntu можно командой:

```
lsb_release -a
```

Общие команды Linux

Вернуться в содержание

Настройка локали в консоли:

1	*sudo shutdown time*	Выключить ОС в момент времени time.
2	*sudo shutdown now*	Выключить ОС прямо сейчас.
3	*sudo reboot*	Перезагрузить ОС сейчас.
4	*logout или exit*	Завершить сеанс работы текущего пользователя.
5	*sudo apt-get update*	Обновить список доступных пакетов
6	*sudo apt-get upgrade*	Обновить доступные пакеты
7	*sudo apt-get install locales*	Установка (обновление) пакета locales.
8	*sudo dpkg-reconfigure locales*	Настройка локализации
9	*sudo dpkg-reconfigure console-setup*	Настройка консоли (выбираем: латинский, славянская кириллица, греческий).
10	*dpkg-reconfigure keyboard-configuration*	Настройка клавиатуры.

Наиболее употребительные команды Linux:

1	*man command-name*	Просмотр справочной системы по команде command-name.
2	*clear*	Очистка экрана консоли.
3	*pwd*	Информация о текущем каталоге.
4	*whoami*	Информация о текущем пользователе.
5	*echo sequence*	Выводит последовательность символов sequence в консоль.
6	*touch file-name*	Создать в текущем каталоге пустой файл с именем file-name
7	*nano file-name*	Редактирование файла file-name в редакторе nano.

8	*cat file*	Просмотреть файл.
9	*tac file*	Просмотреть файл в обратном порядке.
10	*ls* *ls -l*	Просмотр содержимого текущего каталога. Подробный просмотр текущего каталога.
11	*head -n10 file-name*	Вывод 10-ти первых строк файла.
12	*tail -n10 file-name*	Вывод 10-ти последних строк файла.
13	*less file-name*	Удобный постраничный просмотр текстового файла file-name.
14	*wc file-name* *wc -c file-name* *wc -w file-name* *wc -l file-name*	Вывод число строк, слов и символов в файле. Число символов. Число слов. Число строк.
15	*cut -f2,3 file-name* *cut -f3-5 file-name* *cut -f7 -d, file-name* *cut -f1,2 -d ‘ ’ –output-delimiter=$’\t’ file-name*	Вывод 2-го и 3-го столбцов файла. Вывод столбцов файла с 3-го по 5-й. Вывод 7-го столбца. Столбцы разделены разделителем указанным после -d, т.е. запятой. Вывод 1-го и второго столбца, разделённых пробелом. Вывести через разделитель – табуляцию.
16	*tr ‘a’ ‘b’* *tr -s ‘ ‘*	Команда замены символа ‘a’ на ‘b’. Команда удаления повторяющихся пробелов. Данные для команды должны быть получены по конвейеру или перенаправлением ввода-вывода.
17	*grep “xyz” file-name* *grep -r “xyz” dir-name*	Вывод строк файла file-name (файлов каталога dir-name), содержащих последовательность символов «xyz». *Некоторые ключи команды grep:* *-r – рекурсивный поиск в подкаталогах;* *-v – находит все строки, кроме образца поиска;* *-i – регистронезависимый поиск;* *-n – выдаёт также и номер строки.*
18	*sort file-name*	Сортирует строки файла в алфавитном порядке.
19	*uniq file-name*	Удаляет повторяющиеся строки файла.
20	*comm1;comm2*	Последовательное выполнение команд comm1 и comm2.
21	*comm1 && comm2*	Выполнение команды comm2 при условии успешного выполнения команды comm1.
22	*comm1 | comm2*	Вывод команды comm1 перенаправляется на ввод команды comm2.
23	*comm > file* *comm >> file*	Вывод команды comm перенаправляется в текстовый файл file. Если файл существует, он перезаписывается. Вывод команды comm перенаправляется в текстовый файл file. Если файл существует, то вывод команды дописывается в конец файла.
24	*comm 2> file*	Вывод в файл file ошибок команды comm.

	comm 2>> file	Дописывание в файл file ошибок, возникших при выполнении команды comm.

Команды для работы с файлами и каталогами

Вернуться в содержание

4	*cp f1 f2*	Скопировать файл f1 на f2.
5	*mv f1 f2*	Переместить или переименовать файл f1 в f2.
6	*rm file*	Удалить файл.
7	*locate file*	Быстрый поиск файла. Внимание! Пакет надо доустановить командой: *sudo apt install mlocate.*
8	*which program*	Информация о расположении программы (команды).
9	*less file*	Удобный просмотр больших текстовых файлов.
10	*cd*	Перейти в домашний каталог пользователя.
11	*cd dir*	Войти в каталог dir.
12	*cd -*	Вернуться в предыдущий каталог.
13	*mkdir dir*	Создать каталог dir.
14	*rmdir dir*	Удалить каталог dir. Внимание! Данная команда удаляет только пустой каталог.
15	*rm -r dir*	Рекурсивное удаление каталога dir. Удаляет каталог dir и все вложенные каталоги и файлы.
16	*ls dir*	Вывод содержимого каталога dir.
	.	Текущий каталог.
	..	Родительский каталог.
	~	Домашний каталог пользователя.
	Tree [опции] directory	Очень удобный просмотр структуры каталога directory в виде дерева. Основные опции: • -a вывод имён файлов и каталогов; • -d вывод только каталогов; • -h выводит размер файлов; • -o направляет вывод в указанный файл. Замечание! Пакет требует предварительной установки командой*: sudo snap install tree*.
	cp -r dir1 dir2	Рекурсивное копирование каталога dir1 в каталог dir2 со всеми вложенными каталогами и файлами.
	mv -r dir1 dir2	Рекурсивное перемещение (переименование) каталога dir1 в каталог dir2 со всеми вложенными каталогами и файлами.
	Tab	Автодополнение имён файлов и каталогов.

Управление пользователями

Вернуться в содержание

Управление пользователями в Linux:

В Linux существуют три типа пользователей:

- *Администраторы* — привилегированные пользователи с полным доступом к системе. По умолчанию на Linux-сервере после установки операционной системы всегда есть один такой пользователь — root.
- *Локальные* — непривилегированные пользователи. Их учётные записи создаёт администратор. Особенность таких аккаунтов в ограниченном доступе к серверу.
- *Системные* — учётные записи, автоматически создаваемые системой для работы внутренних процессов и служб.

Каждый пользователь имеет свой уникальный идентификатор, UID. Он отличается в зависимости от типа пользователя:

- администратор — 0;
- системный — от 1 до 100;
- обычный — от 100 и выше.

Чтобы упростить процесс настройки прав для новых пользователей, их объединяют в группы. Каждая группа имеет свой набор прав и ограничений. Любой пользователь, создаваемый или добавляемый в такую группу, автоматически их наследует. Если при добавлении пользователя для него не указать группу, то у него будет своя, индивидуальная группа — с его именем. Один пользователь может одновременно входить в несколько групп.

1	`cat /etc/passwd`	Просмотреть информацию о всех пользователях системы.
2	`pinky -l user` `pinky или w`	Просмотреть информацию об отдельном пользователе в удобном (расшифрованном) формате. Просмотреть информацию о всех пользователях, авторизованных в данный момент в системе.
3	`id user`	Узнать ID пользователя.
4	`cat /etc/default/useradd`	Просмотр настроек по умолчанию при добавлении нового пользователя. Эти настройки применяются при использовании команды без параметров вида: useradd any-user.
5	`su user-name`	Быстрое переключение на пользователя с именем user-name, не завершая сеанса.

Узнать текущего пользователя можно выполнив команду:

```
whoami
```

Добавление нового пользователя

Вернуться в содержание

```
useradd [параметры команды] [имя пользователя]
```

Параметры команды:

`-m`	Создаёт указанную домашнюю директорию, если она ещё не существует.
`-d /home/test-user`	Устанавливает /home/test-user в качестве домашней директории.
`-c "Полиграф Шариков"`	Добавляет комментарий.
`-g test`	Указывает группу, в которую попадёт пользователь после создания. Можно использовать с GID или именем группы. Указанная группа должна существовать. Используется в сочетании с ключом -N (отменяет автоматическое создание группы с именем пользователя).
`-G sudo`	Указывает список дополнительных групп. Они перечисляются через запятую без пробелов.
`-s /bin/bash`	Позволяет настроить доступ к shell.
`-r`	Создаёт системного пользователя. Используется, когда вам нужно настроить службу на работу из-под конкретного пользователя. По умолчанию данные таких пользователей не вносятся в /etc/shadow, для них не создаётся домашняя папка.
`-u`	Позволяет указать свой UID, который будет присвоен новому пользователю. В качестве UID указывается положительное целое число. UID должен быть уникален.
`-e 2021-01-01`	Указывает дату, до которой аккаунт будет активен. Дата задаётся в формате YYYY-MM-DD.
`-f 3`	Указывает количество дней до блокировки пользователя, когда его пароль станет недействителен.

Внимание!

Параметр -p меняет пароль при указании после него ХЕША! Сменить или установить пароль для пользователя можно в интерактивном режиме с помощью команды вида:

```
sudo passwd user
```

Пример создания пользователя:

```
useradd -m -u 777 -d /home/users/test-user -c "Тестовый пользователь" -e 2060-01-01 -s /bin/bash test-user
```

Изменение параметров пользователя

Вернуться в содержание

Внимание!

Перед изменением настроек пользователя необходимо убедиться, что он не авторизован в системе и от его имени не запущены процессы. Это можно сделать командами:

- *Просмотр запущенных пользователем процессов:*

```
pgrep -l -u user
```

- *Проверка авторизации пользователя:*

```
pinky user
```

Команда изменения настроек пользователя:

```
usermod [изменяемые параметры] [пользователь]
```

`-m`	Создаёт новую директорию, указанную в качестве домашней (если её не существует), и переносит туда данные из старой.
`-d /home/users/new-test-user`	Меняет домашнюю директорию пользователя на /home/users/new-test-user.
`-c "Чак Норрис"`	Меняет комментарий к пользователю.
`-a -G sudo`	Добавляет пользователя в дополнительные группы.
`-s /bin/bash`	Меняет командную оболочку пользователя.
`-u 100500`	Изменяет UID.
`-e 2100-01-01`	Меняет дату, до которой аккаунт будет активен.

`-f 7`	Меняет количество дней до блокировки пользователя, когда его пароль станет недействителен.
`-l new-test-user`	Меняет имя пользователя на new-test-user.
`-L`	Блокирует аккаунт. Для этого в файле /etc/shadow перед хэшем пароля ставится символ «!».
`-U`	Снимает блокировку с аккаунта (удаляет символ «!» из пароля в /etc/shadow).

Внимание!

Параметр -p меняет пароль при указании после него хеша. Чтобы сменить или установить пароль для пользователя в интерактивном режиме с помощью команды вида:

```
sudo passwd user
```

Удаление пользователей

Вернуться в содержание

Как и в случае с редактированием, перед удалением нужно убедиться, что под ним отсутствуют активные процессы, не редактируются файлы. Иначе существует риск сбоя системы. В программу встроен механизм защиты, поэтому она не позволит удалить пользователя, если он авторизован или под ним работают какие-то службы.

Для удаления пользователей используется команда userdel. Её структура аналогична предыдущим:

```
userdel [что удаляем] [кого удаляем]
```

Основных параметра два:

`-r`	Удаляет папки пользователя: домашнюю директорию, почтовую очередь.
`-f`	Отключает механизм защиты. При использовании этой опции пользователь будет удалён даже при наличии запущенных процессов и пр. Используется на свой страх и риск, так как может привести к сбою системы.

После удаления пользователей важно вручную проверить, что на сервере не осталось файлов или директорий, принадлежащих удалённому пользователю.

Общая информация о группах пользователей

Вернуться в содержание

Группы пользователей в Linux нужны для упрощения администрирования, обеспечения безопасности и управления ресурсами. Они позволяют:

- упростить администрирование, назначая права доступа группе пользователей вместо каждого пользователя отдельно;
- повысить безопасность, ограничивая доступ к важным ресурсам определённым группам пользователей;
- распределять ресурсы между пользователями, создавая группы для совместной работы над проектами.

Просмотреть список групп, в которые входит пользователь, можно командами:

`groups`	Выводит список наименований групп, в которые входит текущий пользователь.
`groups user-name` *или* `id -Gn user-name`	Выводит список групп, в которые входит пользователь с именем user-name.
`id`	Выводит список ID групп, в которые входит текущий пользователь.
`getent group group-name`	Просмотреть список пользователей группы с именем group-name.

Создание группы пользователей

Вернуться в содержание

Просмотреть список всех групп можно в файле:

```
cat /etc/group
```

Чтобы создать группу используется команда groupadd:

```
groupadd new-group
```

Из параметров можно выделить следующие:

`-f`	Если группа с указанным именем или GID уже существует, опция прерывает выполнение команды без соответствующей ошибки.
`-g 100500`	Позволяет назначить свой GID для создаваемой группы.
`-r`	Создаёт системную группу.
`-p p@ssw0rd`	Устанавливает для группы пароль p@ssw0rd. Пароль запрашивается системой при попытке входа в группу с помощью команды newgrp. Не рекомендуется к использованию из-за проблем с безопасностью. Настроенный таким образом пароль можно увидеть в истории команд.

Редактирование групп пользователей

Вернуться в содержание

Для редактирования групп используется команда groupmod. Список изменений задаётся с помощью параметров:

`-g 100500`	меняет GID группы на 100500
`-n another-name`	меняет имя группы на another-name

Например, если нам нужно изменить имя группы test-group на имя named-group, команда будет выглядеть так:

```
groupmod -n named-group test-group
```

Удаление групп пользователей

Вернуться в содержание

Нельзя удалить группу, если она указана в качестве основной для какого-то существующего пользователя. Сначала нужно предварительно удалить этого пользователя из группы.

Само удаление группы выполняется одной командой:

```
groupdel test-group
```

Как и в случае удаления пользователей, нужно вручную проверить, что на сервере не осталось данных, принадлежащих удалённой группе.

Управление пользователями в группе

Вернуться в содержание

Базовым инструментом для управления группами является утилита gpasswd. Она имеет несколько параметров, но с одной особенностью — в отличие от предыдущих примеров, здесь большинство параметров (кроме -A и -M) не сочетаются. То есть в команде может быть только один параметр за раз.

Структура команды проста:

```
gpasswd [опции] [группа]
```

Опции команды:

`-a new-user`	Добавляет new-user в группу.
`-d bad-user`	Удаляет bad-user из группы.
`-A user1,user2,...`	Доступна для использования привилегированным пользователям (с правами root). Назначает список пользователей-администраторов группы.
`-M user1,user2,...`	Доступна для использования привилегированным пользователям. Назначает список участников группы.
`-r`	Отключает пароль группы. После этого только члены группы смогут использовать команду newgrp для подключения к группе.

`-R`	Отключает внешний доступ к группе. После этого только члены группы смогут использовать команду newgrp для подключения к группе.

Как добавить пользователя в группу?

- Если нам потребуется добавить пользователя в новую группу, достаточно будет использовать следующую команду:

```
gpasswd -a new-user test-group
```

- Также для добавления пользователей в новую группу используется описанная выше команда usermod. Следующий пример добавляет пользователя test-user в группу new-group:

```
usermod -a -G new-group test-user
```

- Или, если нужно указать группу new-group в качестве основной группы пользователя test-user:

```
usermod -g new-group test-user
```

Внимание!

Вы не сможете удалить пользователя из его основной группы. Надо сначала добавить его в другую группу и сделать её основной.

- Помимо этого, любой пользователь может сам авторизоваться и добавиться в новую группу с помощью команды:

```
newgrp new-group
```

Эта команда позволяет переключить группу пользователя в рамках текущей сессии, а также автоматически добавляет запрошенную группу в список групп пользователя.

Управление правами доступа к файлам и каталогам

Вернуться в содержание

На каждый файл и каталог можно задать права доступа для следующих пользователей:

1. Для владельца (создателя);
2. Для группы, в которую входит владелец;
3. Для всех остальных пользователей.

Для каждой группы пользователей можно задать следующие права:

1. Чтение – просмотр файла;
2. Запись – разрешение на изменение файла;
3. Выполнение – разрешение на запуск программ и просмотр каталогов.

Получается 3x3 = 9 вариантов:

	Владелец			**Группа владельца**			**Все пользователи**		
	Чтение	**Запись**	**Выполнения**	**Чтение**	**Запись**	**Выполнение**	**Чтение**	**Запись**	**Выполнение**
Символьная запись	r	w	x	r	w	x	r	w	x
Битовая запись	0 | 1	0 | 1	0 | 1	0 | 1	0 | 1	0 | 1	0 | 1	0 | 1	0 | 1
Восьмеричная запись	1-я восьмеричная цифра			2-я восьмеричная цифра			3-я восьмеричная цифра		

Например, строка вида "-r--r-----" состоит из 10-ти символов. Первый символ может быть ‘d’, что является признаком каталога, или символ ‘-’, указывающий, что это НЕ каталог. В двоичном виде данный набор прав можно представить как 100 100 000, в восьмеричном – 440.

Права доступа		
1	*chmod +r file.txt* *chmod -r file.txt*	Установить (снять) разрешение на чтение файла.
2	*chmod +w file.txt* *chmod -w file.txt*	Установить (снять) разрешение на изменение файла.
3	*chmod +x script* *chmod -x script*	Установить (снять) разрешение на выполнение файла или просмотр каталога.
4	*chmod 754 script*	Установить разрешения: 1. Полные права для владельца; 2. Чтение и выполнение для группы владельца; 3. Только чтения для всех остальных.
Смена владельца		
5	*chown user file*	Передать права быть владельцем файла file пользователю user. Бывший владелец может потерять доступ к файлу!
Изменение атрибутов		
6	*chattr +i file* *chattr -i file*	+Запретить (-разрешить) любое изменения, переименование или удаление файла file. Может быть установлено только от имени root!
7	*chattr +a file* *chattr -a file*	+Запретить (-разрешить) только добавление данных в файл file. Данный атрибут может установить только root!
8	*chattr +u file* *chattr -u file*	Файл с атрибутом u при удалении сохраняется на диске для возможности последующего восстановления.

Бывают ситуации, когда определённые программы можно запускать только от имени root. В этом случае можно на ресурс назначить специальные права:

1. SUID (Set User ID root) – право запускать от имени root;
2. SGID (Set Group ID root) – право запускать от имени группы root.

Пример:

```
chmod u+s /usr/sbin/pppd
```

Следует иметь ввиду, что установка таких прав сильно снижает безопасность системы.

Сценарии командной оболочки bash.

Вернуться в содержание

Право выполнения может быть установлено как бинарным откомпилированным программам, так и текстовым файлам сценариев. Принято, что универсальным расширением имени для файла сценария любой командной оболочки Linux является “.sh” (от англ. “shell” - оболочка). Хотя расширение имени в Linux не является обязательным и тип файла определяется по двоичной сигнатуре заголовка файла и по его атрибутам.

Сценарий содержит список команд, каждая из которых начинается с новой строки. Первой строкой сценария необходимо указать путь к исполняющей оболочке:

```
#!/bin/bash

echo Привет участникам естественного отбора!
```

При запуске скрипта необходимо помнить, что для запуска оболочка ищет в ваш файл в специально отведённых для этого каталогах, но не в домашнем каталоге! Поэтому при запуске нужно указать полный путь к этому скрипту:

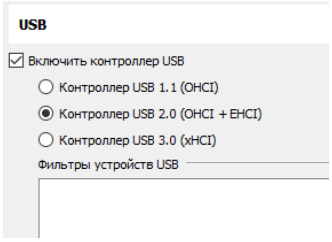
```
$ ~/script.sh
```

~ - это домашний каталог пользователя.

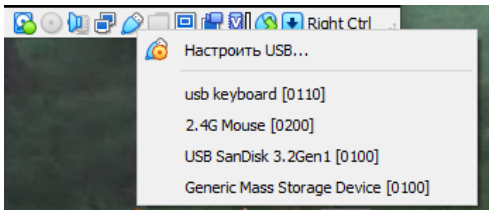
Монтирование блочных устройств

Вернуться в содержание

Если вы желаете монтировать флэшку в VirtualBox, то необходимо до выполнения команд монтирования Linux проверить в настройках виртуальной машины раздел USB:



А также в работающей виртуальной машине надо подключить флэшку, нажав правой кнопкой мыши на значок USB в правой нижней части окна виртуальной машины:



В самой консоли определить имя устройства для монтирования можно, выполнив команду lsblk (list blocks):

```
lsblk
```

Устройство должно появиться в каталоге */dev/*.

Само монтирование выполняем командой вида:

```
sudo mount -t vfat -w /dev/sdb1 /mnt/flash
```

Здесь ключ *-t vfat* указывает на монтирование файловой системы FAT или FAT32, а ключ *-w* разрешает запись в примонтированную флэшку.

Замечание: Каталог /mnt/flash должен быть предварительно создан!

Размонтировать устройство можно, выполнив команду:

```
sudo umount /dev/sdb1
```

или

```
sudo umount /mnt/flash
```

Замечание: Запись на примонтированное устройство осуществляется только с правами sudo (root).

Переменные окружения и псевдонимы

Вернуться в содержание

Переменные окружения создаются командой присваивания:

```
$h=’Hello World’
```

Замечание 1:

Пробелы до и после знака равенства не допускаются!

Замечание 2:

Использование одинарных кавычек (апострофов) экранирует содержимое переменных, т.е. будет выводиться имя переменной:

```
$echo ‘$HOME’
$$HOME
$echo “$HOME”
$/home/user
```

Для создания постоянных переменных, которые будут сохраняться после перезагрузки, необходимо внести их в файл “*~/.bashrc”*.

Обращение к переменной производится через префикс доллара:

```
$echo $h
$Hello World
```

Для сокращения записи часто используемых команд можно использовать псевдонимы:

```
$alias nw=’wc -w’
$nw test.txt
$8
```

Псевдонимы, а также переменные окружения, также могут быть прописаны в файл “*~/.bashrc”* для того, чтобы они сохранялись между сеансами работы.

В Linux можно создавать ссылки на файлы, работа с которыми будет вызывать соответствующие изменения в самом файле. Существует два типа ссылок:

1. Жёсткие – могут указывать только на файл, расположенный в данной файловой системе (но не в примонтированной).
2. Символические – могут указывать на любой файл.

Изменение любых ссылок вызывает аналогичные изменения в самом файле.
Создание ссылок:

1. `ln file.txt link1` – символическая ссылка;
2. `ln -s file.txt link2` – жёсткая ссылка.

Замечание:

При использовании VirtualBox желательно сразу настроить сеть виртуальной машины в режиме «Сеть NAT» (сеть между виртуалками) и добавить в сетевых настройках необходимое количество сетевых адаптеров.

Старые сетевые команды в Linux находятся в пакете *net-tools*, а новые в пакете *iproute2*. По умолчанию в новых дистрибутивах устанавливается только *iproute2*. Эти команды могут потребоваться в работе, поэтому их лучше установить. Чтобы установить *net-tools* нужно выполнить команду:

```
sudo apt install net-tools
```

Здесь:

sudo – запуск команды от имени суперпользователя. Пользователь, вводимый при установке системы, автоматически добавляется в группу *sudo* Другие группы и пользователей в sudo можно добавить командой visudo.

apt – Advanced Package Tool (расширенный инструмент для работы с пакетами Debian/Ubuntu).

Для работы с сетевыми настройками в новых дистрибутивах Linux используется сетевой менеджер командной строки (Network Manager Command Line Interface).

Внимание!

Сетевой менеджер командной строки в Ubuntu Server 20.04 почему-то по умолчанию не установлен.

Проверить наличие и статус Network Manager можно с помощью команды:

```
sudo systemctl status NetworkManager
```

Для установки инструмента nmcli необходимо выполнить команду:

```
sudo apt install network-manager
```

Для запуска менеджера выполняем команду:

```
sudo systemctl start network-manager
```

Замечание:

Для удобства работы можно поставить менеджер Midnight Commander следующей командой:

```
sudo apt install mc
```

Для работы именно с Network Manager (nmcli) необходимо именно его объявить «ответственным» за работу с сетевыми настройками. Для этого надо в файле */etc/netplan/00-installer-config.yaml* закомментировать все строки и прописать следующие настройки:

```
network:
  renderer: NetworkManager
```

После этого необходимо перезапустить систему для того, чтобы изменения вступили в силу.

Замечание:

Для того, чтобы в Ubuntu Server 20.04 изменить имя хоста нужно проделать следующее. Сначала отредактируйте файл /etc/cloud/cloud.cfg и измените параметр «preserve_hostname» с «false» на «true», а затем отредактируйте файл /etc/hostname.

Базовые сетевые настройки в Linux

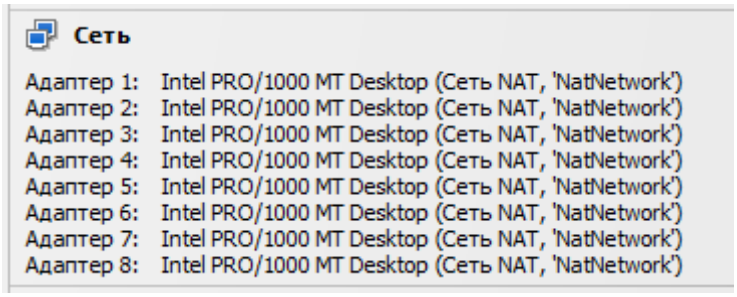
Как в VirtualBox под Windows настроить 8 сетевых адаптеров

Для этого используется утилита из пакета VirtualBox, называемая VBaxManage. Последовательность команд в CMD:

```
1) cd "C:\Program Files\Oracle\VirtualBox"
   – делаем каталог с утилитой текущим
2) VBoxManage list vms
   – смотрим список имеющихся виртуальных машин и находим там UUID нужной машины
3 VBoxManage modifyvm "<UUID>" --nic5 NatNetWork
  VBoxManage modifyvm "<UUID>" --nic6 NatNetWork
  VBoxManage modifyvm "<UUID>" --nic7 NatNetWork
  VBoxManage modifyvm "<UUID>" --nic8 NatNetWork
   - включаем адаптеры с 5-го по восьмой и настраиваем в них «Сеть NAT»
```

Внимание!

Добавленные адаптеры будут присутствовать в виртуальной машине, но не будут отображаться в графическом интерфейсе VirtualBox. Увидеть их можно будет только в сводной информации о настройках машины или в самой ВМ:



Просмотр IP-настроек

Вернуться в содержание

1	`ip link show`	Отображает состояние физических интерфейсов.
2	`sudo ip link set enp0s3 up`	Отключить интерфейс
3	`sudo ip link set enp0s3 down`	Включить интерфейс
4	`sudo ip link set enp0s3 down` `sudo ip link set enp0s3 name eth1` `sudo ip link set enp0s3 up`	Переименовать сетевой интерфейс enp0s3 в eth1
5	`ip address show`	Просмотр сетевых настроек с помощью нового набора команд из iproute2.
6	`ip address show enp0s3`	Просмотр сетевых настроек на интерфейсе enp0s3.
7	`ip -4 address`	Просмотр сетевых настроек для ipv4.
8	`ip route show`	Просмотр таблицы маршрутизации.
9	`hostname`	Вывести имя хоста.
10	`cat /etc/resolv.conf`	Информация о dns-серверах.
11	`cat /etc/netplan/01-network-manager-all.yaml`	Просмотреть все настройки в netplan.

В выдаче команд:

lo – loopback интерфейс,
enp0s3 – сетевой интерфейс,
brd – broadcast, широковещательный адрес данной подсети.

Настройка основных IP-параметров

Вернуться в содержание

Статический IP-адрес

Этот пример устанавливает статический IP-адрес, маску подсети, шлюз по умолчанию и DNS-серверы.

```
network:
  version: 2
  renderer: networkd
  ethernets:
    ИНТЕРФЕЙС:
      dhcp4: false
      dhcp6: false
      addresses: [192.168.1.10/24]        # IP-адрес и маска подсети (CIDR)
      gateway4: 192.168.1.1               # Шлюз по умолчанию
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]     # DNS-серверы (Google Public DNS)
```

- version: 2: Версия спецификации Netplan.
- renderer: networkd: Используемый механизм рендеринга (networkd - стандартный).
- ethernets:: Раздел, определяющий настройки для Ethernet-интерфейсов.
- ИНТЕРФЕЙС:: Имя вашего сетевого интерфейса (например, eth0, ens33).
- dhcp4: no: Отключает DHCP для IPv4.
- dhcp6: no: Отключает DHCP для IPv6.
- addresses: [192.168.1.10/24]: Указывает статический IP-адрес и маску подсети (в формате CIDR). /24 означает маску подсети 255.255.255.0.
- gateway4: 192.168.1.1: Указывает IP-адрес шлюза по умолчанию.
- nameservers: addresses: [8.8.8.8, 8.8.4.4]: Указывает IP-адреса DNS-серверов.

DHCP

Этот пример настраивает интерфейс для получения IP-адреса через DHCP (IPv4). IPv6 отключено.

```
network:
  version: 2
  renderer: networkd
  ethernets:
    ИНТЕРФЕЙС:
      dhcp4: true
      dhcp6: false
```

- dhcp4: yes: Включает DHCP для IPv4.

- dhcp6: no: Отключает DHCP для IPv6.

Статический IP-адрес и несколько DNS-серверов

```
network:
  version: 2
  renderer: networkd
  ethernets:
    ИНТЕРФЕЙС:
      dhcp4: false
      dhcp6: false
      addresses: [192.168.1.10/24]
      gateway4: 192.168.1.1
      nameservers:
        addresses: [1.1.1.1, 1.0.0.1, 8.8.8.8]
```

Этот пример показывает, как указать несколько DNS-серверов. Cloudflare DNS (1.1.1.1, 1.0.0.1) и Google Public DNS (8.8.8.8) используются в качестве примера.

Применение конфигурации Netplan:

После редактирования и сохранения файла конфигурации Netplan, примените изменения с помощью команды:

```
sudo netplan apply
```

Важно: Команда netplan apply может временно прервать сетевое соединение. Если вы работаете удаленно через SSH, будьте осторожны и убедитесь, что конфигурация правильная, чтобы не потерять доступ к серверу.

Проверка конфигурации:

После применения конфигурации, проверьте, что новые настройки вступили в силу. Используйте команду ip address для проверки IP-адреса, ip route для проверки шлюза по умолчанию и resolvectl status для проверки DNS-серверов.

```
ip addr show ИНТЕРФЕЙС
ip route
resolvectl status
```

Работа со временными маршрутами

Вернуться в содержание

Внимание!

По умолчанию Linux не выполняет сквозную маршрутизацию (ip forwarding)!

Для временного, до перезагрузки, но мгновенного запуска маршрутизации надо установить 1 вместо 0 в файле «*/proc/sys/net/ipv4/ip_forward*».

Для постоянной, сохраняющейся после перезагрузки, маршрутизации надо прописать (или раскомментировать) параметр «*net.ipv4.ip_forward=1*» в файле «*/etc/sysctl.conf*».

1	`ip route`	Вывод маршрутной информации. Тут default via (англ. «по умолчанию через») – адрес шлюза по умолчанию, dev – device, устройство, интерфейс.
2	`netstat -rn`	Вывод маршрутной информации командой из устаревшего набора.
3	`ip route add 192.168.2.0/24 via 192.168.100.1`	Добавить временный маршрут на сеть 192.168.2.0/24 через 192.168.100.1
4	`ip route change 192.168.2.0/24 via 192.168.8.10`	Изменить временный маршрут.
5	`ip route append 192.168.2.0/24 via 192.168.8.11`	Добавить второй временный шлюз для сети 192.168.2.0/24 (балансировка маршрута).
6	`ip route delete 192.168.2.0/24 via 192.168.8.11`	Удалить временный маршрут на сеть 192.168.2.0/24 через 192.168.8.12
7	`ip route delete 192.168.2.0/24`	Удалить временный маршрут на сеть 192.168.2.0/24

Работа с постоянными маршрутами

Вернуться в содержание

1		
2		
3		

Диагностика сети в Linux

Связь между IP адресами и MAC с помощью ARP

Вернуться в содержание

1	`arp -a`	Просмотр ARP таблицы
2	`sudo arp -s 192.168.1.2 00:11:22:33:44:55:66`	Добавление статической записи в ARP таблицу.
3	`sudo arp -i enp0s3 -d 192.168.1.2`	Удаление записи из ARP таблицы, соответствующей указанному IP адресу. Параметр -i (interface name) – необязателен.
4	`sudo arp -i enp0s3 10.0.0.1 eth1 pub`	Замаскироваться под указанный IP адрес, чтобы отвечать на предназначенные ему ARP запросы.
5	`$sudo ip link set dev enp0s3 down` `$sudo ip link set dev enp0s3 address 00:11:22:33:44:55` `$sudo ip link set dev enp0s3 up`	Отключить интерфейс. Задать новый MAC адрес. Включить интерфейс.

Сканирование локальных портов и служб

Вернуться в содержание

Для сканирования портов могут применяться команда из старого набора сетевых команд *netstat* и из нового *ss* (socket statistics). Вывод команды ss очень большой и осуществляет перенос строк. Поэтому команду ss целесообразно использовать совместно с командами cat, grep, tr и др.

Основные (наиболее длительные) состояния портов:

- LISTEN - порт прослушивается каким-то процессом в ожидании подключения.
- ESTABLISHED – на порт установлено соединение с удалённой сетевой службой (клиент или сервер).
- TIME_WAIT – Сеанс закрыт, но всё ещё ожидает получения пакетов.

В любом сеансе TCP клиентом считается тот, кто отправил первый запрос (SYN). Каждый порт может прослушиваться только одной службой.

Порты в диапазоне 0-1023 – серверные, т.е. практически никогда не бывают отправителями. Порты в диапазоне 1024-49151 пользовательские, 49152-65535 динамические (частные).

1	`netstat [параметры] | grep 1234` `Типичные наборы параметров:` `-tan` `-na` `-tuan` `-tulpn` `-lntu (прослушиваемые порты)`	Сканирование локальных портов командой из старого пакета net-tools. *Ключи:* *-t (порты TCP);* *-u (порты UDP);* *-a (все порты, прослушиваемые и нет);* *-l (только прослушиваемые порты);* *-n (не выполнять разрешение для задействованных IP-адресов);* *-p (отображать задействованные процессы).*

		1234 – номер порта. Если порт закрыт команда ничего не возвращает.
2	*ss [параметры]* *Пример использования:* *sudo ss -tuap | tr -s ‘ ’ | cut -d ‘ ’ -f 1,2,5,6 --output-delimiter=$’\t’*	Сканирование локальных портов командой из нового набора сетевых команд. Ключи: -t (порты TCP); -u (порты UDP); -a (все порты); -p (отображать процессы).

Управление портами в Linux (Ubuntu)

Вернуться в содержание

Для управления портами в Linux можно использовать **Uncomplicated FireWall** (Простой Брандмауэр). В Linux это команда - ufw.

Примеры использования:

1	*sudo ufw enable*	Запустить Uncomplicated FireWall.
2	*sudo ufw disable*	Остановить Uncomplicated FireWall.
3	*sudo ufw status*	Проверить статус Uncomplicated FireWall.
4	*sudo ufw allow [порт]*	Открыть порт.
5	*sudo ufw deny [порт]*	Закрыть порт.
6	*sudo ufw allow [служба]*	Открыть порт, занимаемый службой (http, ftp, telnet и т.п.)
7	*sudo ufw deny [служба]*	Закрыть порт, занимаемый службой (http, ftp, telnet и т.п.)

Использование утилиты netcat (nc)

Вернуться в содержание

Команда nc (netcat) служит для передачи и получения данных посредством протоколов TCP и UDP. Она имеет большой набор функций, но при этом её достаточно для того, чтобы проверить соединение и провести несложную отладку.

Общий вид команды nc:

```
$nc [параметры] [адрес] [порт(ы)]
```

Параметры:

- -6 – использовать протокол IPv6. По умолчанию используется параметр -4 и IPv4 соответственно;
- -h – вывести справку со списком доступных параметров;
- -i задержка – добавить задержку между отправкой строк или сканированием портов. Задаётся в секундах;
- -l – режим прослушивания. Используется с указанием порта;
- -N – закрыть соединение при достижении конца файла при его отправке;
- -n – Работать с IP-адресами напрямую, не задействуя DNS, также отключить поиск портов;
- -P имя-пользователя – указать имя пользователя для подключения к прокси;
- -x адрес:порт – указать адрес и порт для подключения к прокси;
- -p порт – указать номер порта. В большинстве случаев порт считывается без указания параметра;
- -U – использовать сокет домена UNIX (для межпроцессного взаимодействия);
- -u – использовать протокол UDP, по умолчанию используется TCP;
- -v – подробный режим. Используется при сканировании портов;
- -W количество-пакетов – закрыть соединение после получения определённого количества пакетов;
- -w таймер – включить таймер для ограничения времени соединения. Задаётся в секундах;
- -z – отключить отправку данных. Используется при сканировании портов.

1	`nc -vz 192.168.1.2 1234`	Просканировать порт 1234 указанного адреса.
2	`nc -vz 192.168.1.2 1-1000 2>&1 | grep succeeded`	Просканировать порты с 1 по 1000 указанного адреса, перенаправив стандартный поток ошибок (2) стандартное устройство вывода (консоль, т.е. - 1).
3	`nc -vzu 192.168.31.247 1-1000 2>&1 | grep succeeded`	Тоже самое, но с включением портов UDP (они всегда открыты).
4	`nc -nlv 1234`	Прослушивание порта 1234 на локальном хосте.
5	`На ПК1: nc -lp 1234` `На ПК2: nc 192.168.1.3 1234`	Использование Net Cat для организации простейшего чата. *Замечание:* *Порт должен быть открыт. Например, командой «sudo ufw allow 1234».*
6	`На ПК1: nc -lp 1234 > f1.txt` `На ПК2: nc 192.168.1.3 < f1.txt`	Использование Net Cat для организации пересылки файла f1.txt с ПК2 на ПК1. *Замечание:*

		Сначала надо открыть сессию на принимающей стороне, а затем уже на отправляющей!

Сканер сетевых портов и удалённых служб Net Mapper

Вернуться в содержание

nmap (Network Mapper) — это мощный инструмент для сетевого сканирования, используемый для обнаружения устройств, определения открытых портов, анализа операционных систем и выявления уязвимостей в сети. Благодаря своей гибкости и обширному набору функций, Nmap является незаменимым инструментом для специалистов по кибербезопасности, администраторов сетей и исследователей. Он позволяет получать информацию о сетевой инфраструктуре, проводить диагностику сетевых проблем и тестировать защиту от вторжений.

По умолчанию утилита не установлена в Ubuntu. Установить nmap можно командой:

```
sudo apt install nmap
```

Общий синтаксис команды

Nmap использует простой и логичный синтаксис:

```
nmap [опции] <цель>
```

Цель: IP-адрес, имя хоста, диапазон IP или подсеть.
Опции: Флаги, определяющие типы сканирования, вывод и настройки производительности.

Форматы адресов

Nmap поддерживает различные форматы ввода IP-адресов:

- Одиночный IP: 192.168.1.1

- Диапазон IP: 192.168.1.1-254
- Подсеть (CIDR): 192.168.1.0/24

Форматы портов

Вы можете сканировать порты, используя различные форматы:

- Одиночный порт: -p 80
- Диапазон портов: -p 20-80
- По именам сервисов: -p http,ftp

Примеры команд:

1	*nmap scanme.nmap.org*	Протестировать специальный тестовый ресурс для nmap.
2	*nmap 192.168.1.1*	Сканирование одиночного хоста.
3	*nmap 192.168.1.1-100*	Сканирование диапазона хостов.
4	*nmap 192.168.1.0/24*	Сканирование подсети CIDR.
5	*nmap -iL hosts.txt*	Сканирование по списку хостов из файла.
6	*nmap -iR 10*	Сканирование случайных хостов.
7	*nmap 192.168.1.0/24 --exclude 192.168.1.5*	Исключение хоста из диапазона сканирования.
8	*nmap -sS 192.168.1.1*	TCP SYN (быстрое сканирование)
9	*nmap -sT 192.168.1.1*	TCP CONNECT (медленное сканирование)
10	*nmap -sU 192.168.1.1*	UDP сканирование.
11	*nmap -p- 192.168.1.1*	Сканирование всех портов.
12	*nmap -p 80,443 192.168.1.1*	Сканирование указанных портов (443, 80).
13	*nmap -sV 192.168.1.1*	Версионная детекция. Показывает ПО и его версии.
14	*nmap -T4 192.168.1.1*	Установка скорости тестирования. От самой медленной T1 до самой быстрой T5.
15	*nmap --mtu 32 192.168.1.1*	Изменение размеров mtu для обхода брандмауэров.
16	*nmap -O 192.168.1.1*	Определение операционной системы.
17	*nmap -sP 192.168.1.1*	Ping-сканирование узла.
18	*nmap -p 443 --open 192.168.1.0/24*	Сканирование в указанной подсети порта 443 (https) с выводом только открытых портов.

** Цветом выделены наиболее часто используемые команды*

Брандмауэр Linux

Традиционное приложение iptables

iptables (IP Tables) - приложение брандмауэра Linux, сменившее в 2001 году предыдущее приложение ipchains, появившееся в Linux в 1999 году. В настоящее время происходит переход на новое приложение nft, однако iptables будет ещё долгое время использоваться, т.к. на нём по-прежнему работают целые ранее созданные продукты и системы автоматизации.

Приложение iptables, как и всё прочее в Linux, использует текстовые файлы, организованные в виде **таблиц (tables)**:

1. **filter** – таблица правил фильтрации пакетов;
2. **NAT** – таблица трансляции сетевых адресов в передаваемых пакетах;
3. **mangle** (калечить, кромсать) – позволяет вносить изменения в передаваемые пакеты.

Таблицы состоят из наборов правил – цепочек (chains). **Цепочки таблицы filter**:

1. **INPUT** – контролирует пакеты, входящие в узел;
2. **OUTPUT** – обрабатывает пакеты, исходящие из узла;
3. **FORWARD** – обрабатывает пакеты пересылаемые на другой узел.

Результатом работы правил является одно из действий (целей):

1. **ACCEPT –** пропустить пакет;
2. **DROP –** отбросить пакет;
3. **RETURN –** вернуть пакет предыдущей цепочке.
4. **LOG –** записать в журнал регистрации.

Формат команды iptables:

```
$iptables -t <таблица> <действие> <цепочка> <дополнительные_параметры>
```

Если не указана <таблица>, то по умолчанию работа ведётся с таблицей filter!

Наиболее важные действия:

1. **-L** (list) – вывести список правил в цепочке или все цепочки;
2. **-A** (append) – добавить правило в конец цепочки;

3. **-I** (insert) – вставить правило в заданную позицию цепочки;
4. **-D** (delete) – удалить указанное правило;
5. **-P** (politics) – изменить политики (действия по умолчанию) в цепочке;

Дополнительные параметры:

1. **-i** (interface) – задаёт интерфейс;
2. **-p** (protocol) – задаёт используемый протокол;
3. **-s** (source) – задаёт URL или IP-адрес источника пакетов;
4. **-d** (destination) – задаёт URL или IP-адрес получателя пакетов;
5. **-dport** (destination port) – порт TCP/UDP;
6. **-j** (jump) – цель (ACCESS, DROP, RETURN)

Примеры команд настройки брандмауэра iptables:

1	*sudo iptables -A INPUT -i ens33 -p tcp -s 1.2.3.0/24 --dport 22 -j ACCEPT*	Добавить правило в цепочку INPUT на интерфейс ens33 по протоколу TCP для порта 22 (SSH), предписывающее пропускать пакеты из сети 1.2.3.0/24.
2	*sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT*	Добавить правило в цепочку INPUT на порт по протоколу TCP для порта 443 (HTTPS), предписывающее пропускать все входящие пакеты.
3	*sudo iptables -I INPUT 2 -I ens33 -p tcp -s 1.2.3.5 --dport 443 DROP*	Вставить во 2-ю позицию цепочки INPUT на интерфейс ens33 по протоколу TCP для порта 443 (HTTPS) правило, предписывающее отбрасывать все пакеты от узла 1.2.3.5.
4	*sudo iptables -I INPUT 2 -i ens33 -p tcp --dport 22 -j DROP*	Вставить во 2-ю позицию цепочки INPUT на интерфейс ens33 по протоколу TCP для порта 22 (telnet) правило, предписывающее отбрасывать все пакеты.
5	*sudo iptables -t filter -L INPUT –line-numbers*	Вывести все правила таблицы filter цепочки INPUT с номерами строк.
6	*sudo iptables -D INPUT 5*	Удалить 5-е правило из цепочки INPUT.
7	*sudo iptables -A INPUT -j LOG*	Регистрировать в журнале все срабатывания правил из цепочки INPUT.
8	*sudo iptables -A INPUT -s 192.168.122.0/24 -j LOG*	Регистрировать в журнале срабатывания правил для входящих пакетов из сети 192.168.122.0/24.
9	*sudo iptables -A INPUT -i ens33 -p tcp -s 1.2.3.0/24 --dport 22 -j ACCEPT -m comment --comment «Разрешить доступ администратору»*	Добавить правило в цепочку INPUT на интерфейс ens33 по протоколу TCP на порт 22 (telnet) из сети 1.2.3.0/24 с соответствующим комментарием.

		Используется для создания самодокументируемого кода.
10	*$iptables -A OUTPUT -p tcp -d twitter.com --dport 443 -j DROP* *$ iptables -A OUTPUT -p tcp -d twitter.com --dport 443 -j DROP*	Запретить конкретный веб-сайт (Twitter).
11	*sudo iptables -P INPUT DROP*	Изменить политику, согласно которой все входящие пакеты по умолчанию будет отбрасываться.
12	*sudo iptables -P OUTPUT ACCEPT*	Изменить политику, согласно которой все исходящие пакеты по умолчанию будут пропускаться.
13	*sudo iptables-save*	Сохранить все сделанные изменения в таблицах iptables.
14	*sudo iptables-restore*	Отменить сделанные изменения в iptables.

Последним правилом в каждой цепочки записывается правило, определяющее действие по умолчанию. Если оно не задано, то все пакеты, не подпавшие ни под одно из правил, пропускаются (ACCEPT).

Новое приложение nft

Вернуться в содержание

+++++++