

#bash #QEMU #ipcalc #ifconfig #NetTools #ip #iplink #sysctl #bridge #nano

Дополнение статьи: https://vk.com/wall-153221588_7609

Сохранение настроек сетевого моста при перезагрузке.

Для автоматической настройки сетевого моста после перезагрузки нужна утилита bridge-utils.

Без неё сетевой мост необходимо будет каждый раз настраивать заново, иначе даже соединения с интернетом или роутером не будет.

А именно: указать главный адаптер, подключить сетевой мост, и указать маршруты по умолчанию.

```
$ sudo ip link set br0 up
$ sudo ip link set enp3s0 master br0
$ sudo bridge link
$ sudo route add 192.168.0.120 via 192.168.0.1
$ sudo route add default gw 192.168.0.1 br0
```

```
$ sudo route del default gw 192.168.0.1 br0
$ sudo route del 192.168.0.120 via 192.168.0.1
```

В Archlinux утилиту можно установить так.

```
$ sudo pacman -S bridge-utils --noconfirm
```

В Debian так.

```
$ sudo su
$ echo 'deb http://ftp.de.debian.org/debian sid main' > /etc/apt/sources.list.d/ftp.de.debian.org.list
$ sudo apt update
$ sudo apt install bridge-utils -y
```

Команды утилиты.

```
# Создать мост.
$ brctl addbr bridge_name
# Добавьте устройство к мосту, например eth0:
$ brctl addif bridge_name eth0
# Показать текущие мосты и к каким интерфейсам они подключены:
$ brctl show
# Настроить мостовое устройство:
$ ip link set dev bridge_name up
# Чтобы удалить мост, вам нужно сначала установить его в положение down:
$ ip link set dev bridge_name down
$ brctl delbr bridge_name
```

Сохранение настроек Network Manager.

Если у вас сетевой менеджер Network Manager, то с сохранением настроек моста при перезагрузке могут возникнуть проблемы. Чтобы это исправить будем вручную управлять настройками всех адаптеров и соединений сетевого менеджера.

Для этого отредактируйте файл конфигурации сетевого менеджера «/etc/NetworkManager/NetworkManager.conf» или создайте отдельную конфигурацию по пути «/etc/NetworkManager/conf.d/».

```
$ sudo nano /etc/NetworkManager/NetworkManager.conf
# или
$ sudo mkdir -p /etc/NetworkManager/conf.d/
$ sudo nano /etc/NetworkManager/conf.d/managed.conf
# Далее просто пропишите секцию и её настройку.
[ifupdown]
managed=true
```

Тем самым мы говорим, чтобы менеджер брал настройки из файла «/etc/network/interfaces».

Также, по умолчанию менеджер каждый раз по активации любого соединения переписывает настройки файла «/etc/resolv.conf». Чтобы это предотвратить и управлять dns-настройками вручную просто создайте конфигурацию по тому же пути: «/etc/NetworkManager/conf.d/dns.conf».

```
$ sudo nano /etc/NetworkManager/conf.d/dns.conf
[main]
dns=none
```

Не забудьте включить или отключить resolvconf.service в system:

```
$ systemctl start resolvconf.service
$ systemctl enable resolvconf.service
# или
$ systemctl stop resolvconf.service
$ systemctl disable resolvconf.service
```

Теперь настройте сетевые интерфейсы «/etc/network/interfaces». Например, адрес моста пусть будет 192.168.0.120, адрес роутера по умолчанию 192.168.0.1, маска 24.

Не забудьте про обязательный loopback интерфейс и автоматические или статические адреса всех ваших сетевых интерфейсов, иначе они даже не запустятся. Включая WiFi интерфейс, например, wlp0s3. У меня в виртуальной машине WiFi нет.

```
$ sudo nano /etc/network/interfaces
```

```
auto lo
iface lo inet loopback
```

```
allow-hotplug enp0s3
auto enp0s3
# iface enp0s3 inet dhcp
```

```
allow-hotplug br0
auto br0
iface br0 inet static
# iface br0 inet dhcp
```

```

address 192.168.0.120
netmask 255.255.255.0
broadcast 192.168.0.255
gateway 192.168.0.1
# dns-nameservers 192.168.0.1 8.8.8.8
# pre-up ifconfig br0 hw ether xx:xx:xx:xx:xx:xx
# pre-up ip link set br0 address xx:xx:xx:xx:xx:xx
bridge_ports eth0
bridge_stp off
bridge_fd 0
bridge_maxwait 0
post-up route add 192.168.0.120 via 192.168.0.1
post-up route add default gw 192.168.0.1 br0
post-down route del default gw 192.168.0.1 br0
post-down route del 192.168.0.120 via 192.168.0.1

```

Строка, которая изменяет МАК-адрес не обязательна, поэтому она закомментирована. Она также указана, как в варианте настройки при помощи утилиты `ifconfig`, так и при помощи стандартной утилиты `ip`. Широковещательный адрес рекомендую всё-таки указывать, хоть он и не обязателен. Шлюз указывать обязательно. DNS сервер не обязателен. команды маршрутизации обязательны.

Описание команд конфигурационного файла.

`address address` - адрес.

`netmask netmask` - маска сети.

`broadcast` - широковещательный_адрес.

`network network_address` - адрес сети.

`metric metric` - Метрика(целое число).

`gateway address` - Шлюз по умолчанию.

`pointopoint адрес` - Адрес удалённой точки.

`media type` - Тип носителя, зависящий от драйвера.

`hwaddress class address` - Аппаратный адрес. Класс - это одно из следующих значений:

`ether`, `ax25`, `ARCnet` или `netrom`. Адрес зависит от выбранного класса.

`mtu size` - Размер MTU.

Описание команд `pre-up`, `post-up`, `pre-up`, `post-down`.

- `pre-up` Запустить команду до поднятия интерфейса.
- `post-up` Запустить команду после поднятия интерфейса.
- `pre-down` Запустить команду перед отключением интерфейса.
- `post-down` Запустить команду после отключения интерфейса.

Сохранение настроек Connman.

Если у вас сетевой менеджер `connman`, то сохранить настройки при перезагрузке здесь будет немного посложнее.

Для установки в Debian воспользуйтесь следующими командами.

```
$ sudo su
```

```
$ echo "deb http://ftp.de.debian.org/debian sid main" >>
```

```
/etc/apt/sources.list.d/ftp.de.debian.org.list
```

```
$ apt update
```

```
$ sudo apt install connman connman-gtk connman-ui -y
```

Для установки в Archlinux:

```
$ sudo pacman -S connman --noconfirm
```

Затем внесите изменения в агент и активируйте сервис.

```
$ sudo connmanctl
```

```
agent on
```

```
quit
```

```
$ sudo systemctl enable connman
```

Чтобы понять какие настройки примутся после перезагрузки я вносил изменения при помощи графической утилиты «connman-gtk». Настройки находятся примерно здесь: «/var/lib/connman/ethernet_080027a0c313_cable/settings». На каждом ПК путь будет меняться.

Различные варианты настроек и команд можете посмотреть [здесь](#). Самая первая конфигурация – это копия после сохранения из графической утилиты.

Сохранение настроек netctl.

А вот здесь всё гораздо проще и сложнее одновременно.

Сложнее в том, что утилита нормально устанавливается только в Archlinux. Впихнуть её в Debian и другие системы, вроде Fedora мне так и не удалось. Однако, на основе кода из Archlinux я написал свой собственный [Makefile](#) для скачивания её из реп-я arch-a и установки в систему. Предусмотрел также и сборку утилиты для упаковки в тот или иной установщик. Имеется возможность установки в любой каталог, а также полного контроля над путями всех скриптов и шаблонов утилиты.

Работу утилиты не тестировал, но установки и де-инсталляции (удаления) проходят корректно в любой ОС Linux.

Настройки и команды утилиты можно посмотреть [здесь](#).

Ну а сегодня на этом всё. Надеюсь я хоть немного вас заинтересовал.

Спасибо за внимание. Всем Удачи, до новых встреч, Пока-Пока!