Основные сетевые команды в Linux

Команды для работы с компьютерной сетью – это неотъемлемый инструмент для любого специалиста. Сориентироваться среди многочисленных команд и документации может быть довольно сложно, и иметь единую точку справки играет важную роль при выполнении задач, связанных с сетью.

Эта статья содержит список из 20 важных сетевых команд для Linux.

Предварительный требования

- Доступ к командной строке.
- Учётная запись администратора с правами **sudo**.

20 сетевых команд для Linux

В Linux имеется множество полезных сетевых команд и инструментов. Эти команды обычно предназначены для выполнения сложных сетевых задач, таких как мониторинг, устранение неполадок и настройка сети. Большинство утилит для работы с сетью включены в состав старого пакета net-tools или более современного iproute2.

> В большинстве дистрибутивов Linux доступны как команды net-tools, так и iproute2. Однако рекомендуется использовать инструменты iproute2 из-за их гибкости и скорости.
>
> Несмотря на то, что net-tools считаются устаревшими, они до сих пор широко используются в старых скриптах и конфигурациях.
>
> *NETWORK ADMIN*

Синтаксис конкретной команды может варьироваться в зависимости от версии команды. Предварительно проверьте синтаксис команды с помощью следующей команды:

```
man [command]
```

Команда man выводит на экран руководство для указанной команды в терминале.

Ниже представлена краткая сводка из 20 сетевых команд для Linux.

ip

Команда ip представляет собой интегрированный инструмент для работы с сетями в операционных системах Linux. Она облегчает просмотр и настройку маршрутизации, интерфейсов, сетевых устройств и туннелей.

Эта команда входит в состав пакета iproute2 и заменяет множество старых инструментов для работы с сетью, таких как команды route, ifconfig и netstat.

Синтакстис:

```
ip [options] object [command]
```

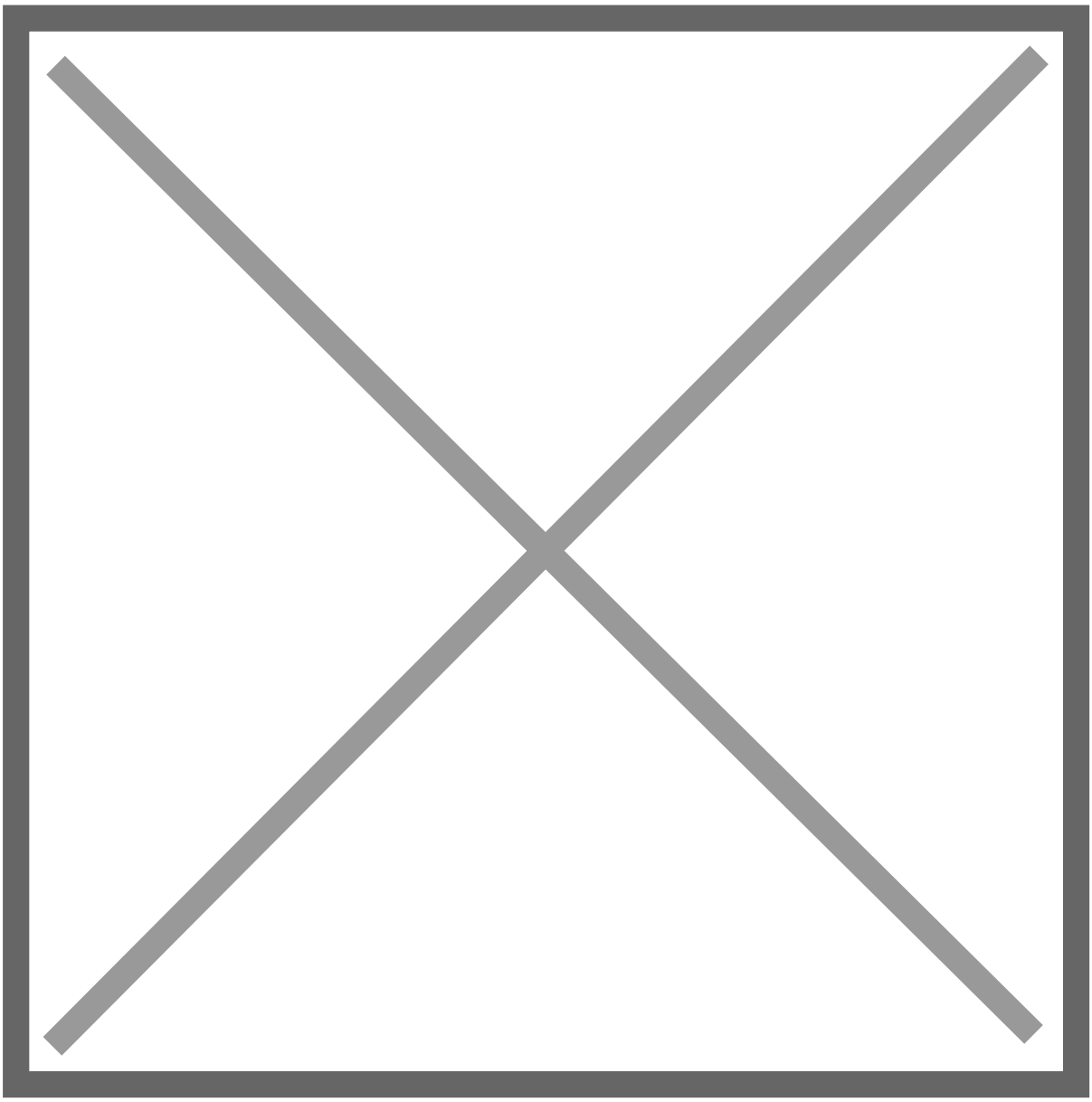
Каждый компонент команды выполняет следующие функции:

- **options** – это параметры командной строки, которые влияют на поведение команды.
- **object** – отображает объекты, которые могут быть настроены.
- **command** – представляет собой подкоманду, действие, которое выполняется с объектом. Доступные команды зависят от конкретного объекта.

Пример:

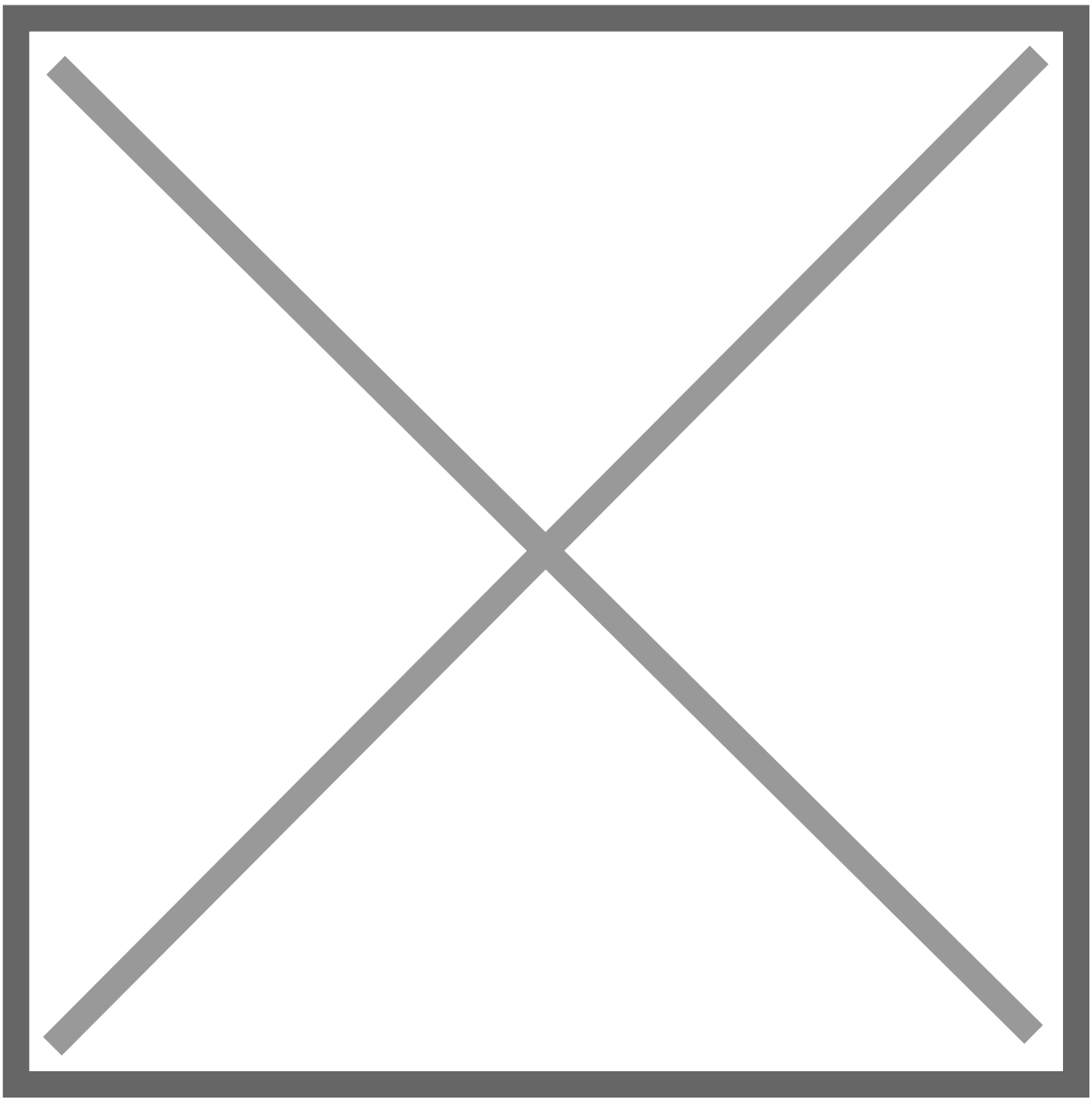
При использовании команды ip без параметров, объектов или команд, отображается меню справки.

```
ip
```



Для просмотра текущей версии, используйте опцию -V.

```
ip -V
```



Результат вывода содержит информацию о версии пакета и библиотеки, используемых утилитой ip.

ip addr

Команда ip addr позволяет управлять и просматривать IP-адреса сетевых интерфейсов. Ее альтернативными названиями являются ip address или ip a.

Синтаксис:

```
ip addr [subcommand]
```

Список доступных подкоманд для данного объекта:

- **add** – Осуществляет добавление нового адреса.
- **show** – Показывает адреса протокола.

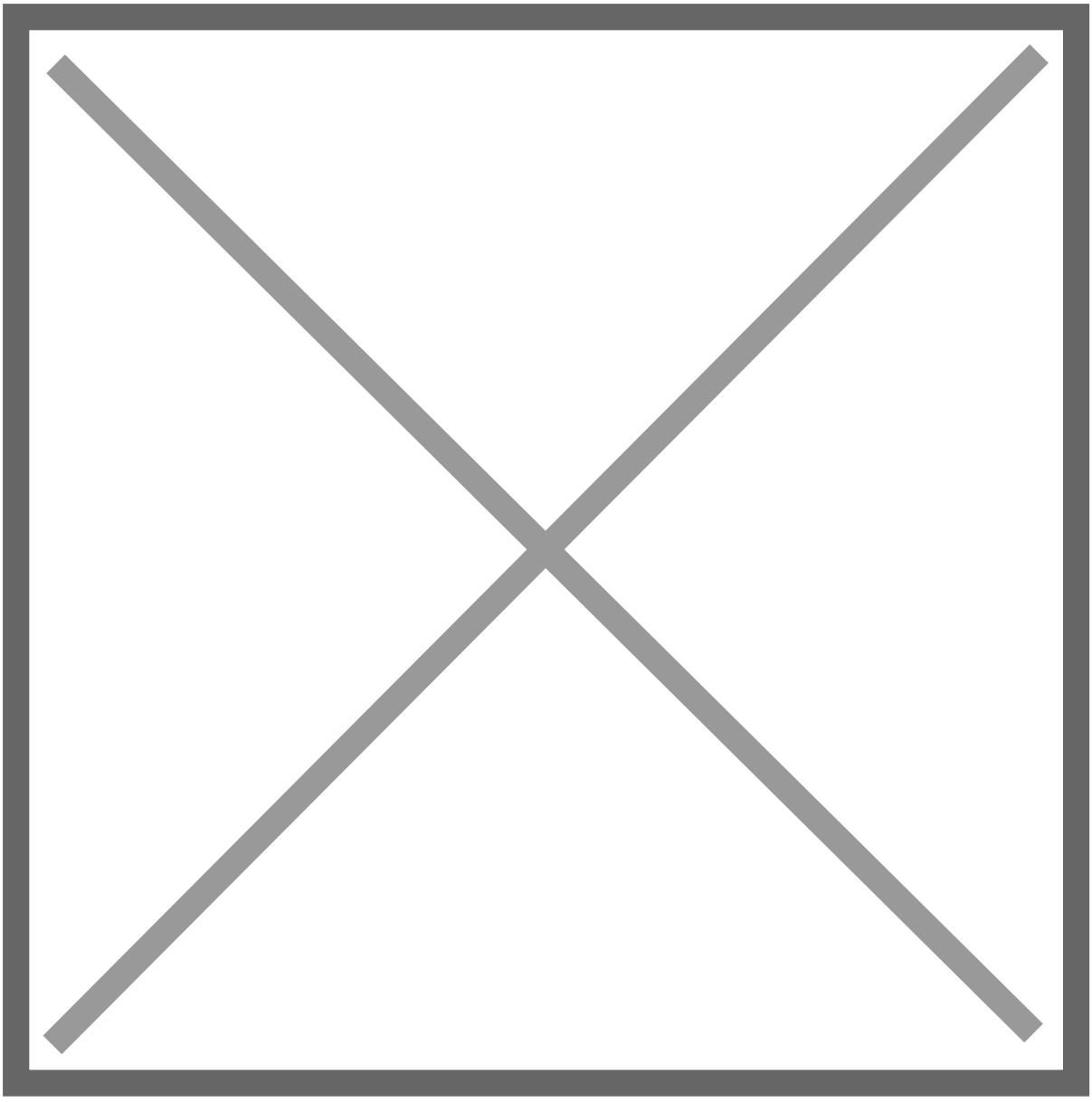
- **del** – Осуществляет удаление адреса.
- **flush** – Удаляет адреса в зависимости от заданных критериев.

У каждой подкоманды есть дополнительные параметры и ключевые слова, которые позволяют выполнять конкретные задачи с адресами сетевого интерфейса.

Пример:

Команда ip addr без каких-либо подкоманд показывает информацию о сетевом интерфейсе, включая соответствующие IP-адреса:

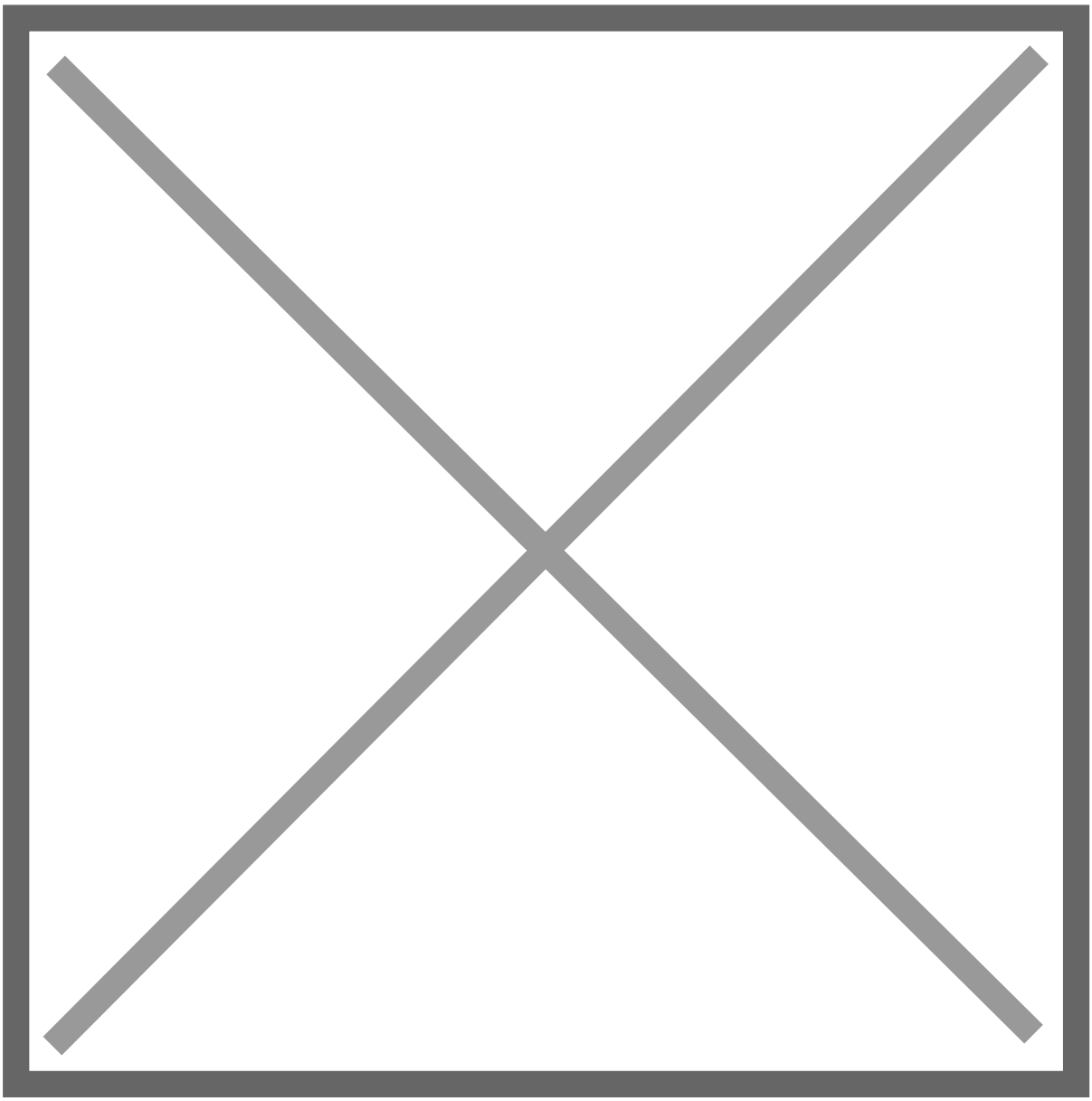
```
ip addr
```



Результат для команды ip addr show будет такой же.

Для просмотра информации о конкретном сетевом интерфейсе, выполните команду ip addr show с указанием его имени.

```
ip addr show [interface]
```



Команда осуществляет фильтрацию вывода команды ip addr, отображая только информацию, связанную с указанным интерфейсом.

ip link

Команда ip link предоставляет возможность управления и отображения информации о сетевых интерфейсах, позволяя просматривать, изменять, включать и выключать их.

Синтаксис:

```
ip link [subcommand] [options] [interfaces]
```

С помощью подкоманд можно выполнять следующие операции:

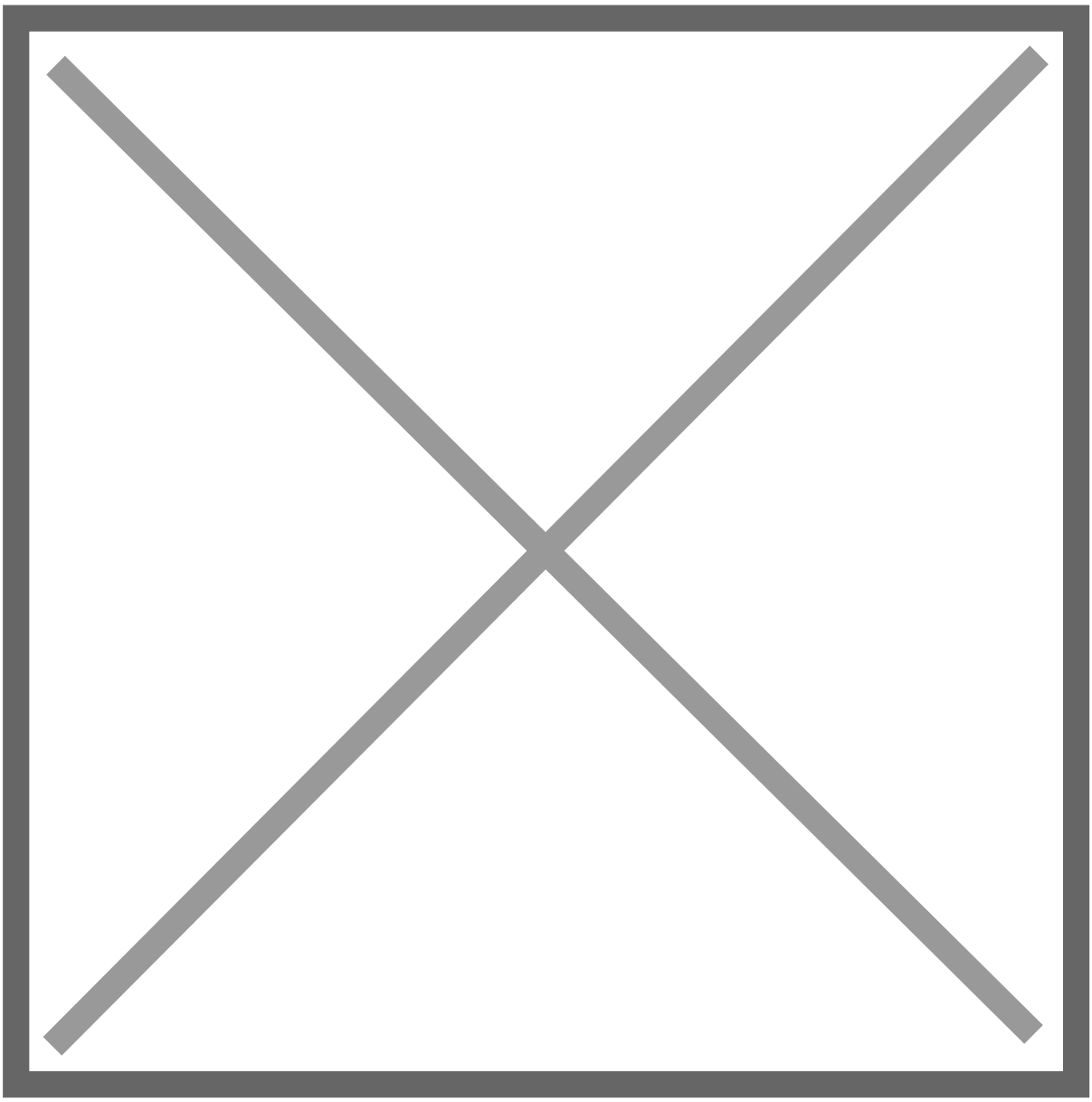
- **show** – Показывает информацию о сетевом интерфейсе.
- **set** – Вносит изменения или добавляет информацию к сетевому интерфейсу.
- **add** – Создает новый сетевой интерфейс.
- **del** – Осуществляет удаление сетевого интерфейса.

Подкоманды предоставляют дополнительные опции и позволяют выбирать определенные интерфейсы.

Пример:

Команда ip link без дополнительных подкоманд и параметров выводит информацию о всех сетевых интерфейсах.

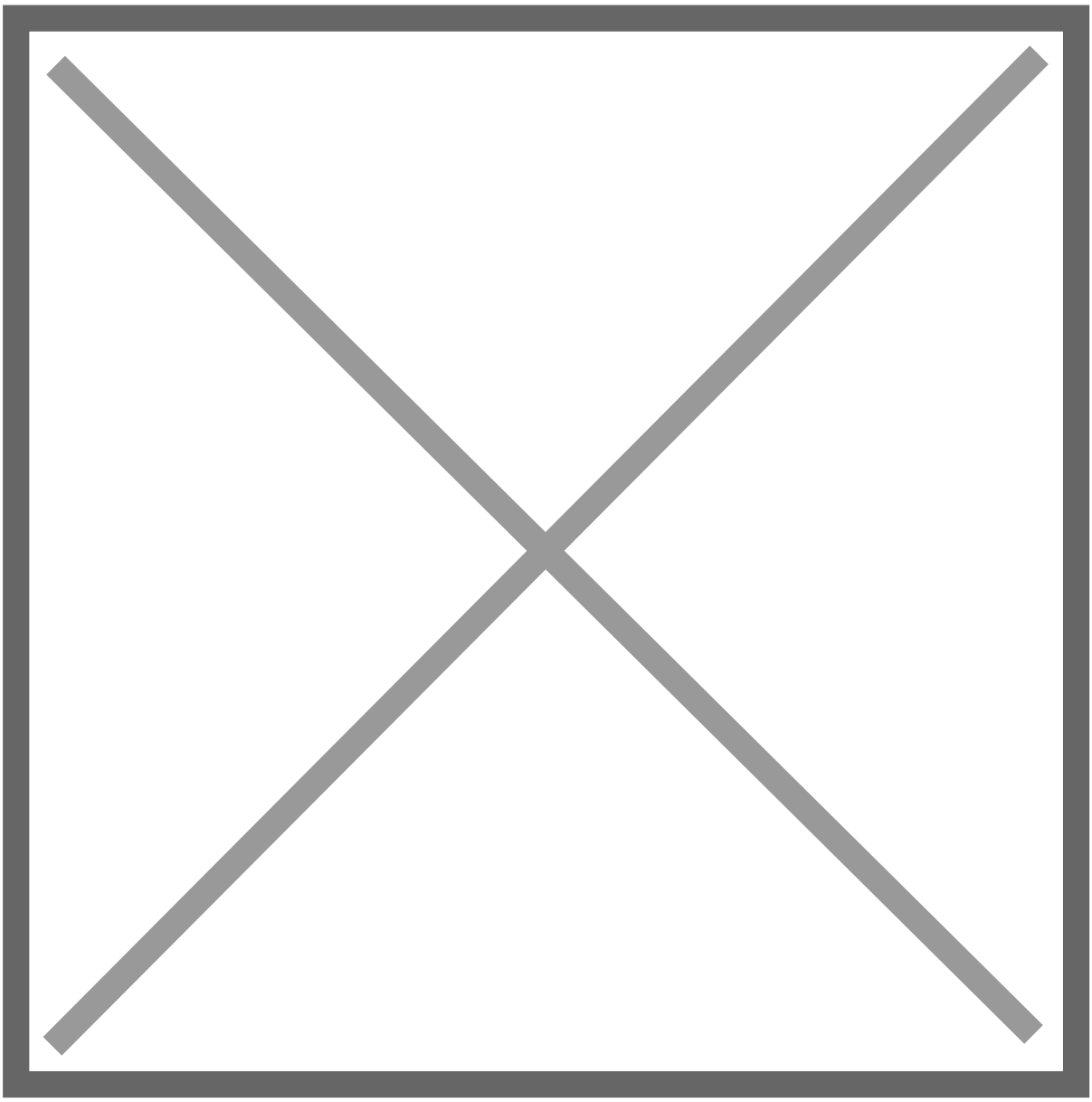
```
ip link
```



Команда ip link show выдает идентичный вывод.

Для выключения интерфейса, выполните следующую команду в роли суперпользователя:

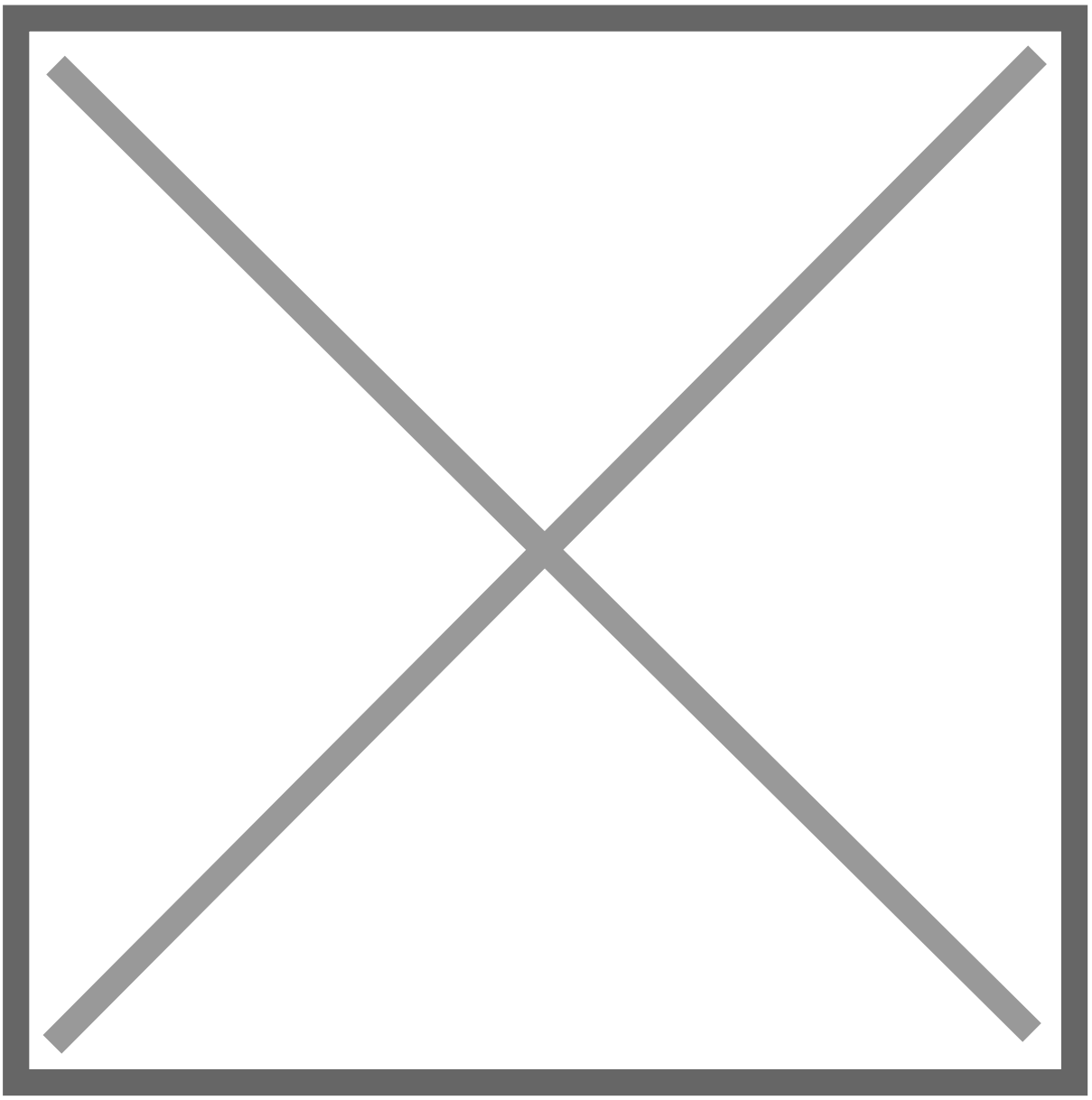
```
sudo ip link set [interface] down
ip link show
```



После выполнения команды состояние интерфейса меняется на DOWN.

Чтобы активировать интерфейс, примените ключевое слово up:

```
sudo ip link set [interface] up
```



Состояние интерфейса переходит в режим UP.

ip route

Команда ip route предоставляет доступ к таблице IP-маршрутизации и позволяет пользователям настраивать её и выполнять другие важные сетевые задачи, связанные с маршрутизацией.

Синтаксис:

```
ip route [subcommand] [options] [destination]
```

В качестве подкоманд доступны следующие действия:

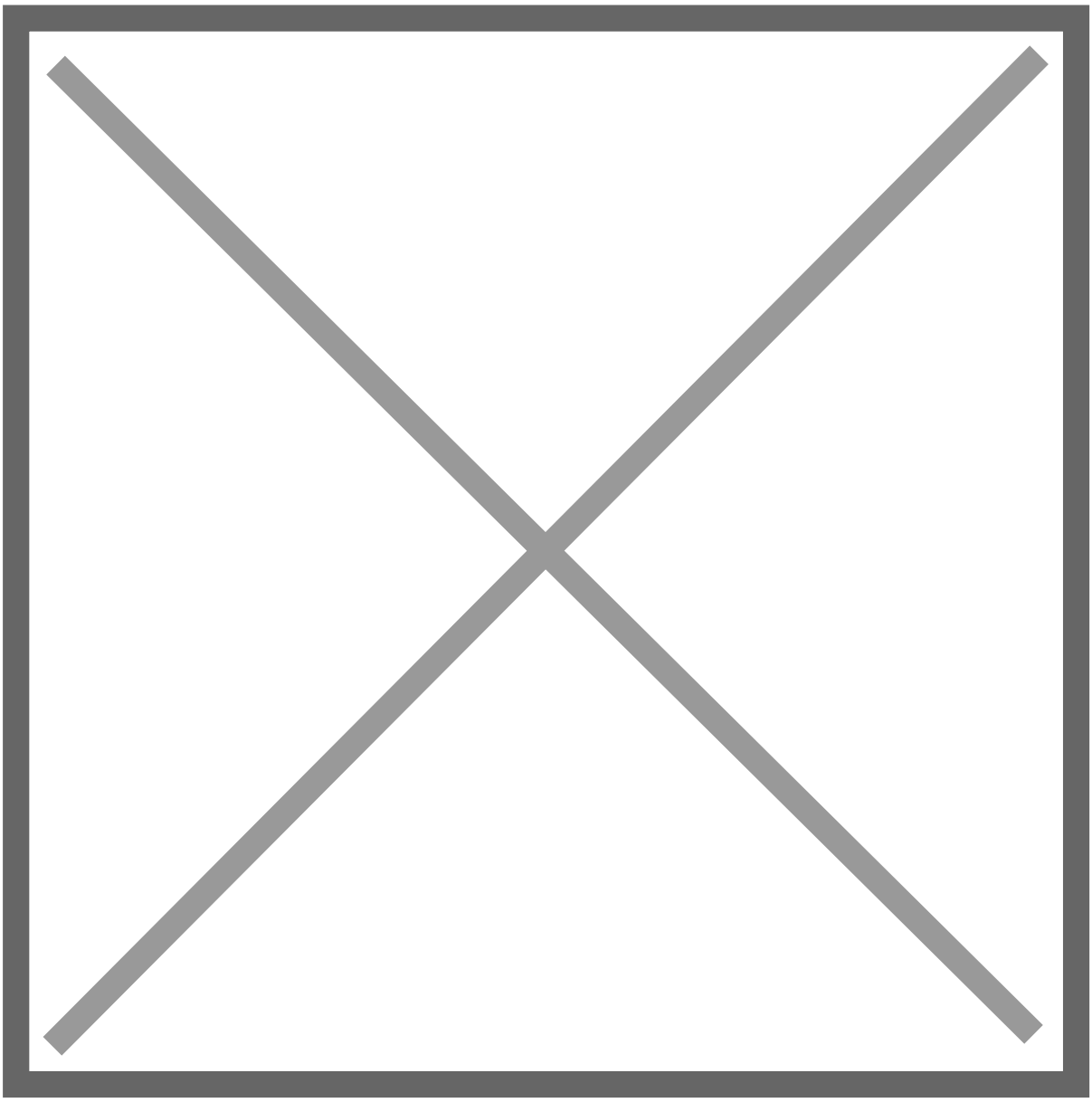
- **show** – Показывает таблицу маршрутизации.
- **add** – Осуществляет добавление нового маршрута в таблицу.

- **del** – Осуществляет удаление маршрута из таблицы.
- **change** – Вносит изменения в существующий маршрут.

Параметр [destination] указывает, куда направляется сетевой трафик. Дополнительные опции обеспечивают дополнительное управление потоком трафика.

Пример:

```
ip route show
```



Каждая строка в выводе соответствует отдельным маршрутам в таблице.

ifconfig

Команда ifconfig (конфигурация интерфейса) управляет и предоставляет информацию о сетевых интерфейсах в системе. Эта команда является частью пакета net-tools.”

Несмотря на ограниченные функции по сравнению с командой ip, команда ifconfig всё еще активно применяется для настройки сетевых интерфейсов.

Синтаксис:

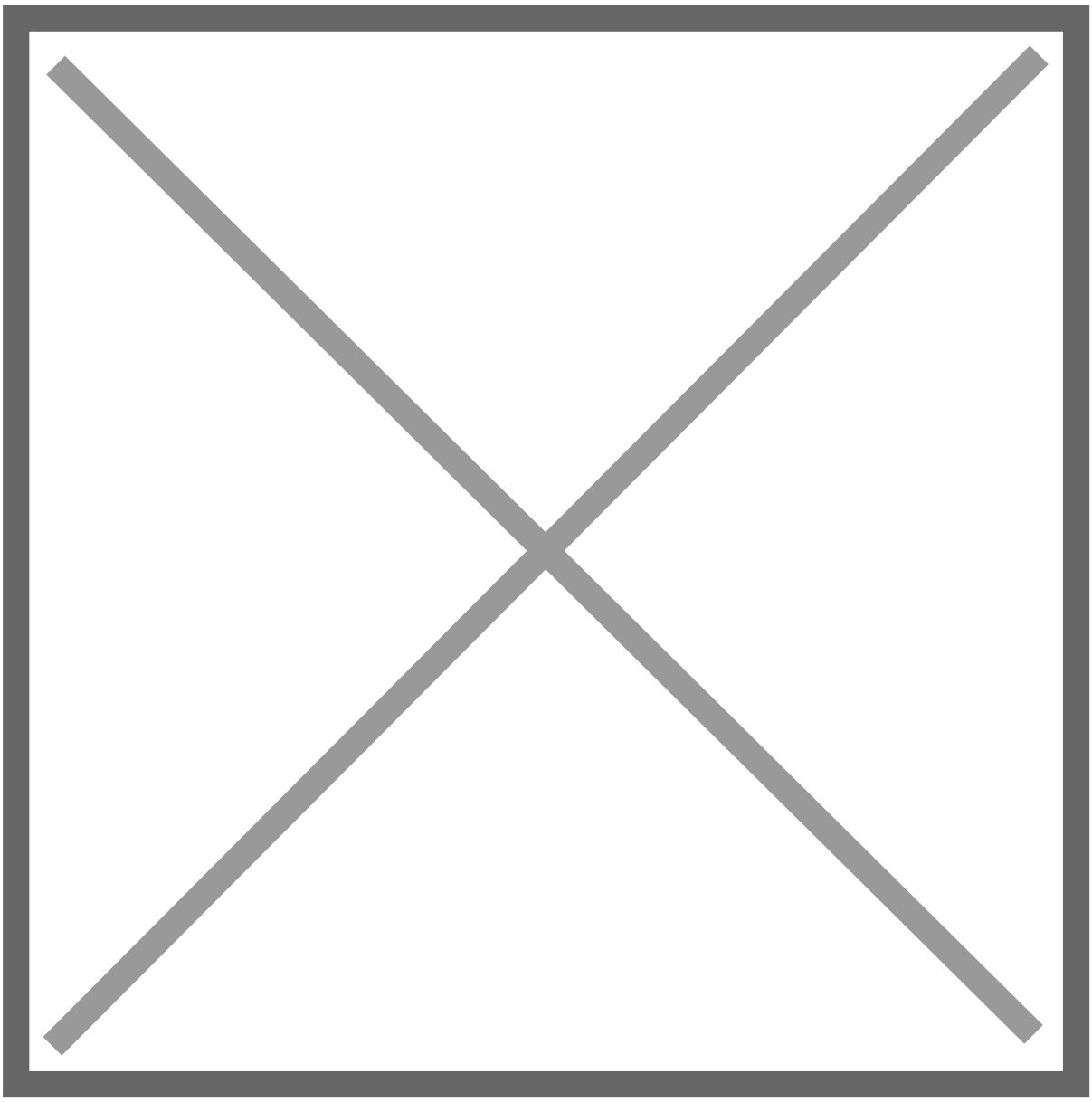
```
ifconfig [interface] [options]
```

Структура синтаксиса представляется следующим образом:

- **interface** – Сетевой интерфейс, для которого требуется настройка или отображение информации. Параметр не является обязательным, и если интерфейс не указан, то будет показан статус всех активных интерфейсов.
- **options** – Опции командной строки для выполнения конкретных действий или настройки определенных параметров. Также важно отметить, что параметр является необязательным.

Пример:

```
ifconfig -s
```



Эта команда предоставляет краткую сводку с важной информацией о текущих интерфейсах.

dig

Команда dig предназначена для выполнения запросов в систему доменных имен (DNS) и поиска информации о DNS-записях. Она собирает информацию о доменных именах и связанных записях.

Для устранения проблем с DNS и проверки настроек DNS в системе Linux используйте команду dig. Она отлично подходит для создания сценариев и автоматизации задач, связанных с анализом сети. Эта мощная команда настолько популярна в решении сетевых проблем, что существует версия dig для Windows.

Синтаксис:

```
dig [options] [domain] [record type] [DNS server]
```

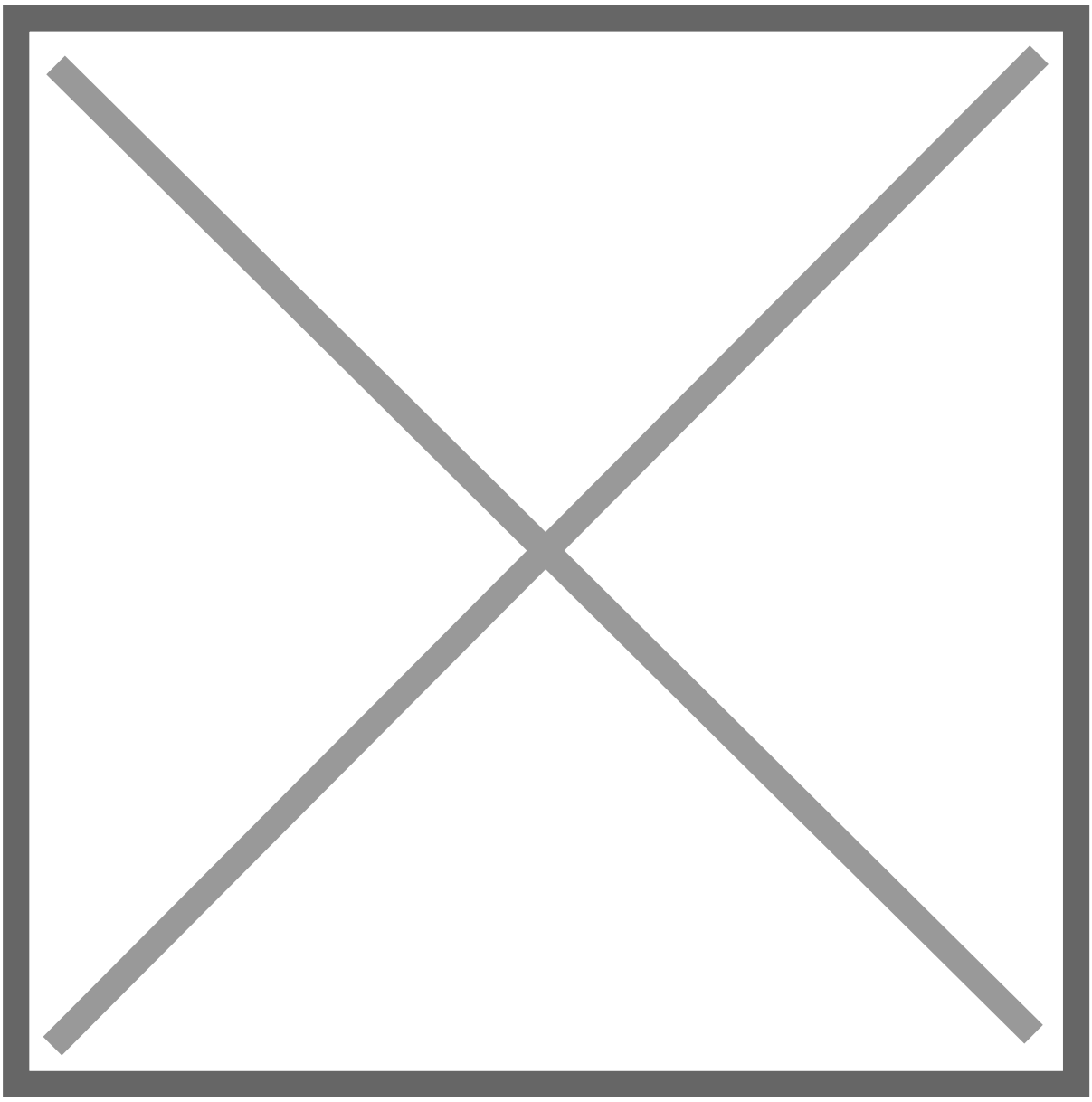
Элементы команды включают:

- **options** – Аргументы, влияющие на функционирование команды.
- **domain** – Имя домена, которое требуется запросить.
- **record type** – Тип DNS-записи для запроса. Значение по умолчанию – записи типа A.
- **DNS Server** – Заданный DNS-сервер для выполнения запроса.

Все параметры необязательны. По умолчанию команда выводит информацию о DNS-разрешителе и статистику запросов без дополнительных опций.

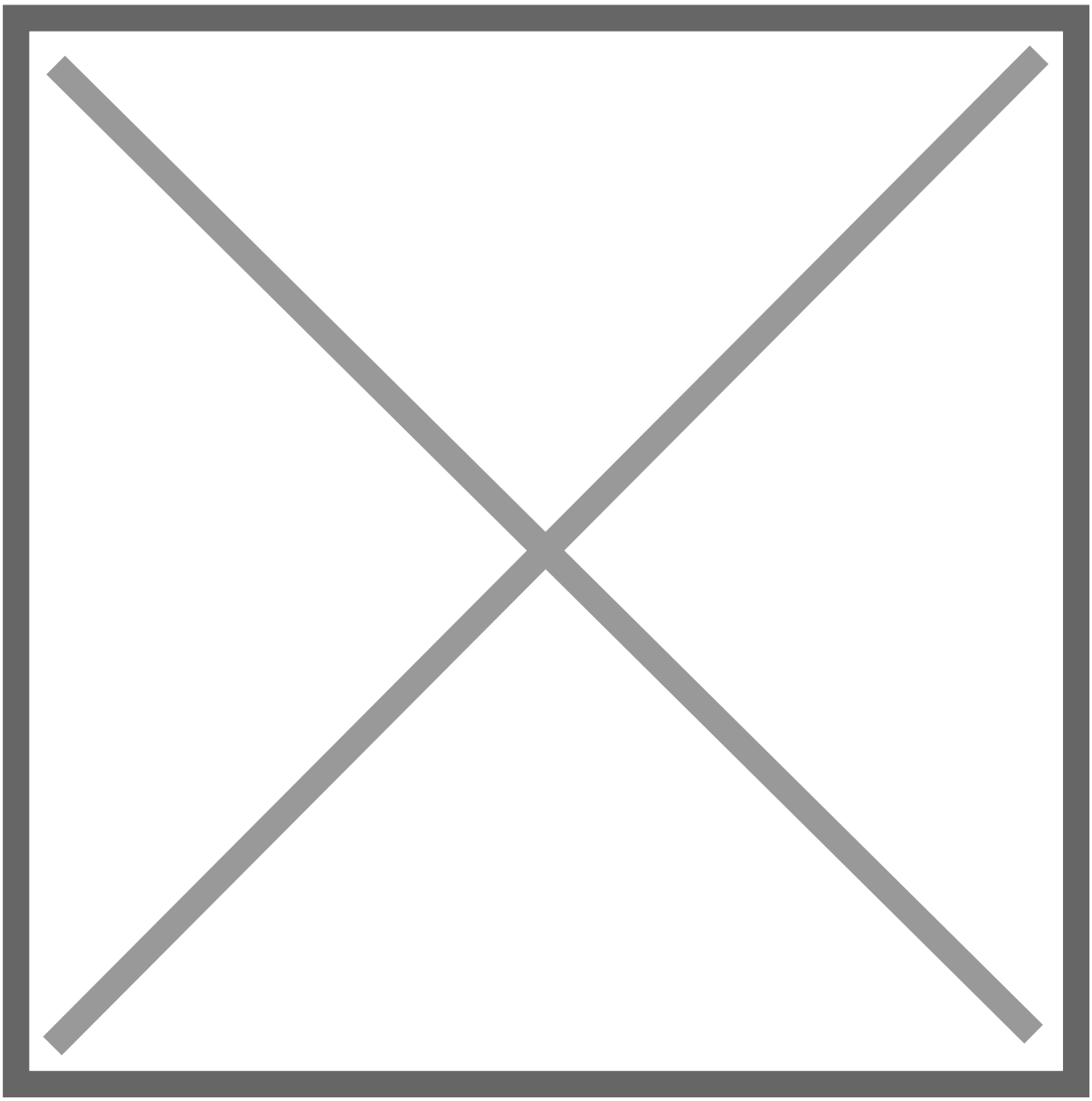
Пример:

```
dig google.com
```



Вместо этого, можно предоставить IP-адрес и использовать опцию -x для выполнения обратного DNS-поиска.

```
dig -x 8.8.8.8
```



В разделе ANSWER SECTION в выводе отображается запрошенное доменное имя.

nslookup

Команда nslookup схожа с командой dig. Основное отличие между этими командами заключается в том, что nslookup предоставляет интерактивный режим. Он позволяет проводить диагностику и выполнять запросы к DNS-серверам, что полезно при решении задач сетевого анализа и DNS.

Команда поддерживается на большинстве операционных систем, включая Unix-подобные и Windows.

Синтаксис:

```
nslookup [domain] [DNS server]
```

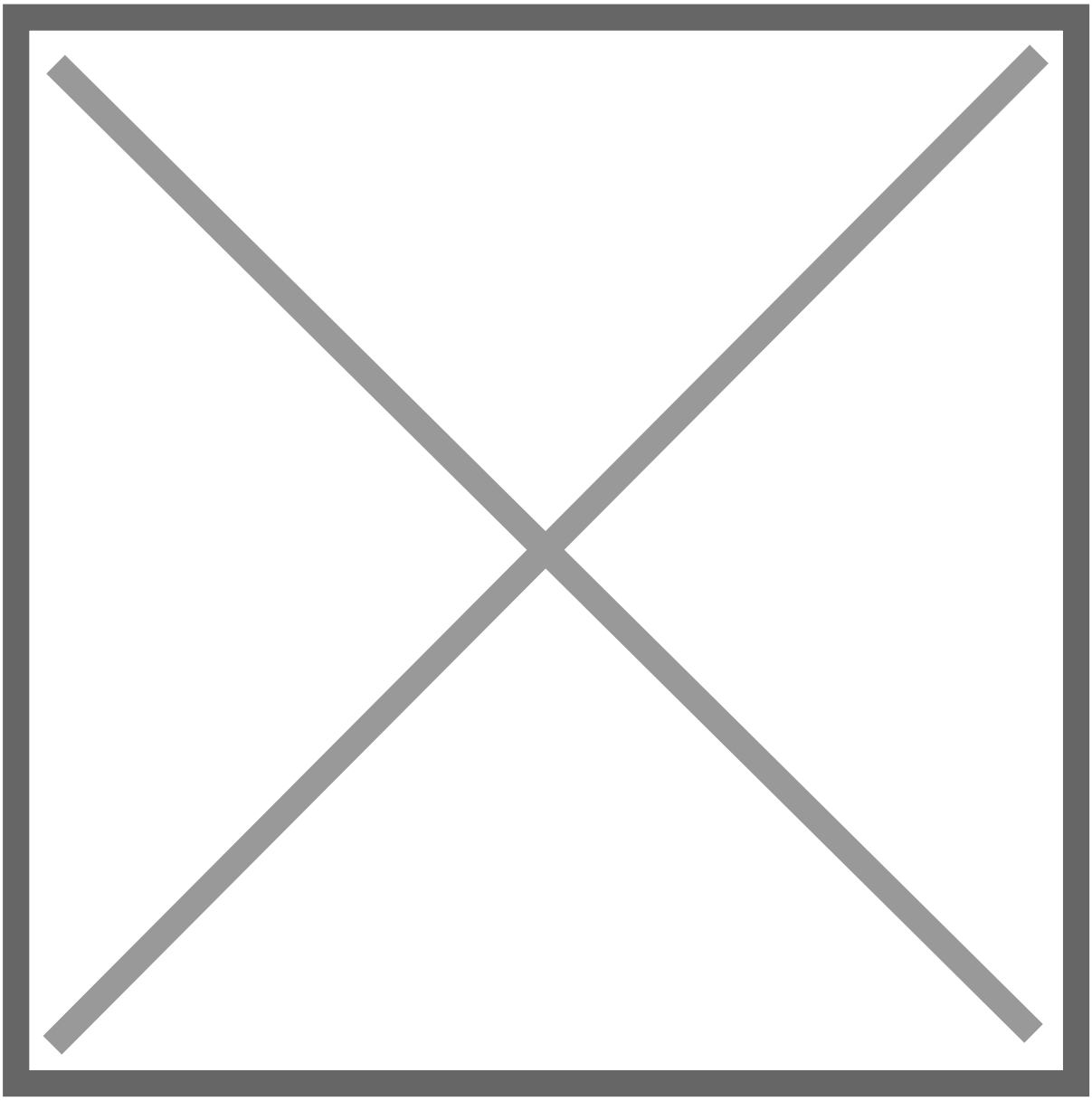
Элементы команды включают:

- **domain** – Имя домена для поиска. Если не указывать имя, то в интерактивном режиме можно выполнять запросы к нескольким доменам.
- **DNS server** – DNS-сервер, используемый для поиска. По умолчанию используется DNS-сервер системы, если не указан конкретный.

По умолчанию запрос выполняет поиск доменных записей типа A.

Пример:

```
nslookup google.com
```



Результат выводит информацию о DNS-разрешении для предоставленного домена.

netstat

Команда netstat (сетевая статистика) представляет собой сетевую утилиту, отображающую различные сетевые статистические данные. Она предоставляет информацию о статусе сетевых портов и их доступности.

Команда входит в состав пакета net-tools и считается устаревшей. Рекомендуется использовать команду ss, которая входит в состав iproute2. Другие функциональности команды netstat доступны с помощью команды ip.

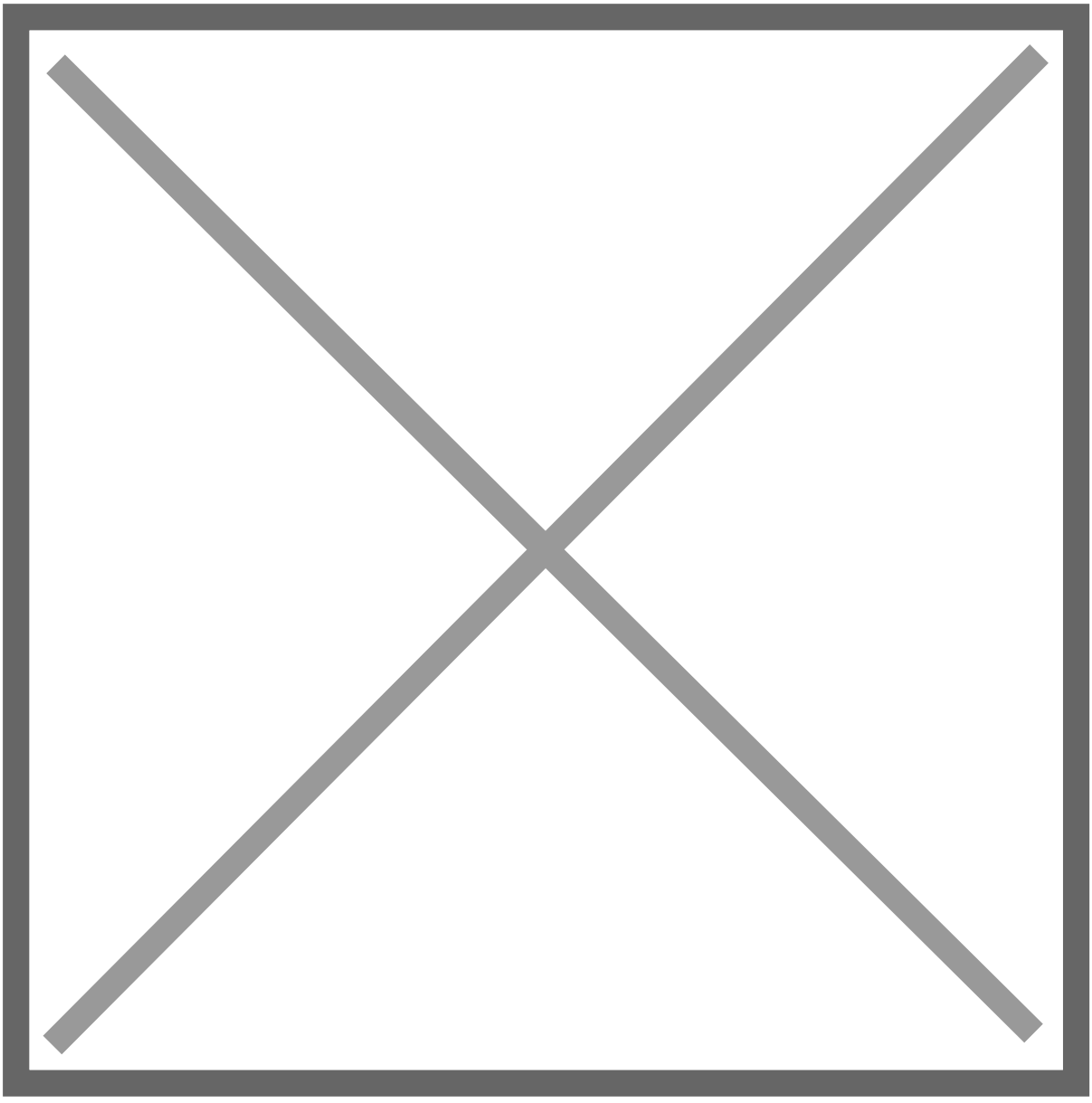
Синтаксис:

```
netstat [options]
```

С помощью команды можно комбинировать различные опции, чтобы настроить вывод и отобразить конкретные типы сетевой информации. Без дополнительных опций команда перечисляет открытые сокеты для всех настроенных семейств адресов.

Пример:

```
netstat -at
```



Результат отображает все активные TCP-соединения в системе.

traceroute

Команда traceroute – это инструмент сетевой диагностики, который поддерживается в Linux, macOS и Windows. С её помощью можно отслеживать маршрут, по которому пакеты идут к заданной цели в сети TCP/IP.

С помощью этой команды можно обнаруживать проблемы с маршрутизацией и узкими местами, отображая промежуточные узлы, через которые проходят пакеты при перемещении от отправителя к получателю.

По умолчанию выполнение трассировки составляет 30 прыжков с размером пакета 60 байтов для IPv4 (80 байтов для IPv6).

Синтаксис:

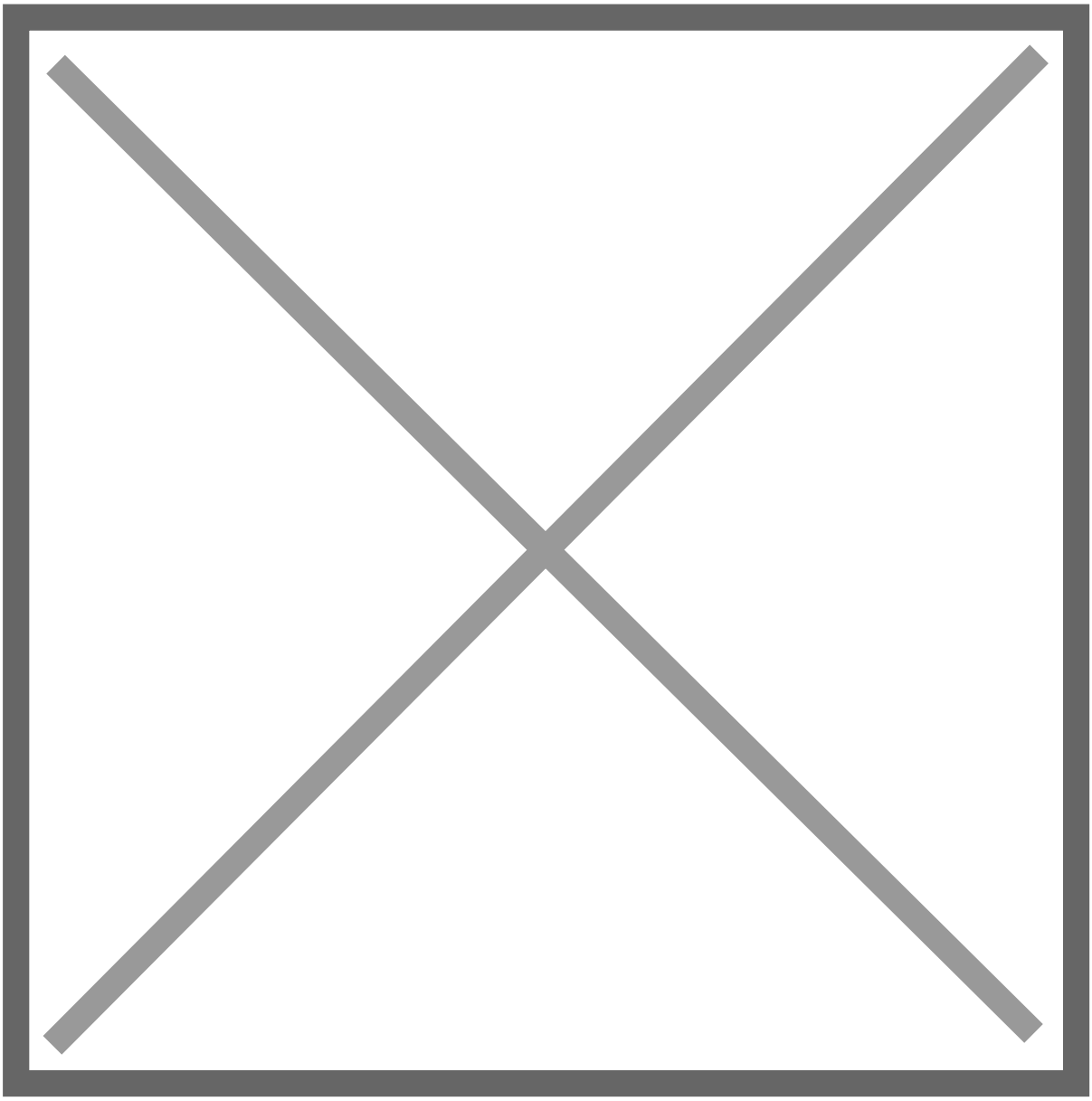
```
traceroute [options] [hostname/IP]
```

Для работы требуется параметр [hostname/IP], а дополнительные опции определяют, выполнять ли DNS-запросы, значение TTL и тип пакета.

Пример:

Для отслеживания маршрута пакета с использованием протокола TCP выполните команду traceroute от имени администратора с опцией -T.

```
sudo traceroute -T 184.95.56.34
```



Результат показывает последовательный маршрут от источника до места назначения.

tracepath

Команда `tracepath` аналогична команде `traceroute`. Она выявляет маршруты и задержки от отправителя до получателя, отображая маршрутизаторы и сетевые переходы.

Несмотря на то, что `traceroute` – известная команда с обширными опциями, `tracepath` представляет собой простой инструмент для отображения сети, доступный на большинстве систем Linux. Для более подробной информации, сравните `tracepath` и `traceroute`.

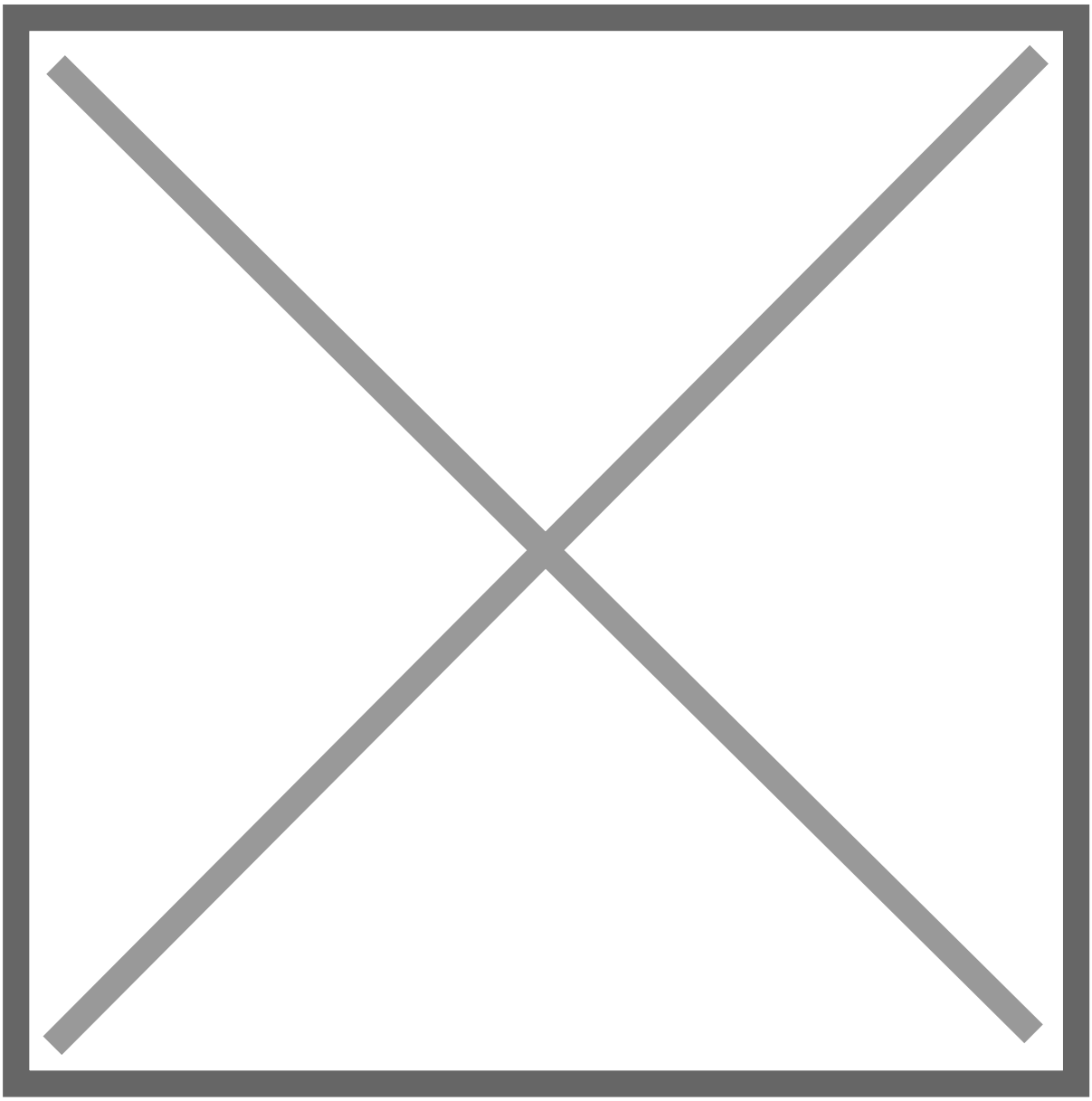
Синтаксис:

```
tracepath [options] [hostname/IP]
```

Дополнительные [options] определяют поведение запроса, включая количество переходов и опцию выполнения обратного DNS-поиска для адресов. Поле [hostname/IP] обязательно и указывает место назначения.

Пример:

```
tracepath [hostname/IP]
```



В результатах видно номер шага, IP-адрес или определенное имя хоста и время задержки туда и обратно (RTT) для каждого шага.

host

Команда `host` – это простой инструмент для проведения DNS-запросов. Она выполняет разрешение IP-адресов в доменные имена и наоборот.

Используйте эту команду для выполнения запросов DNS и базового поиска и устранения проблем DNS.

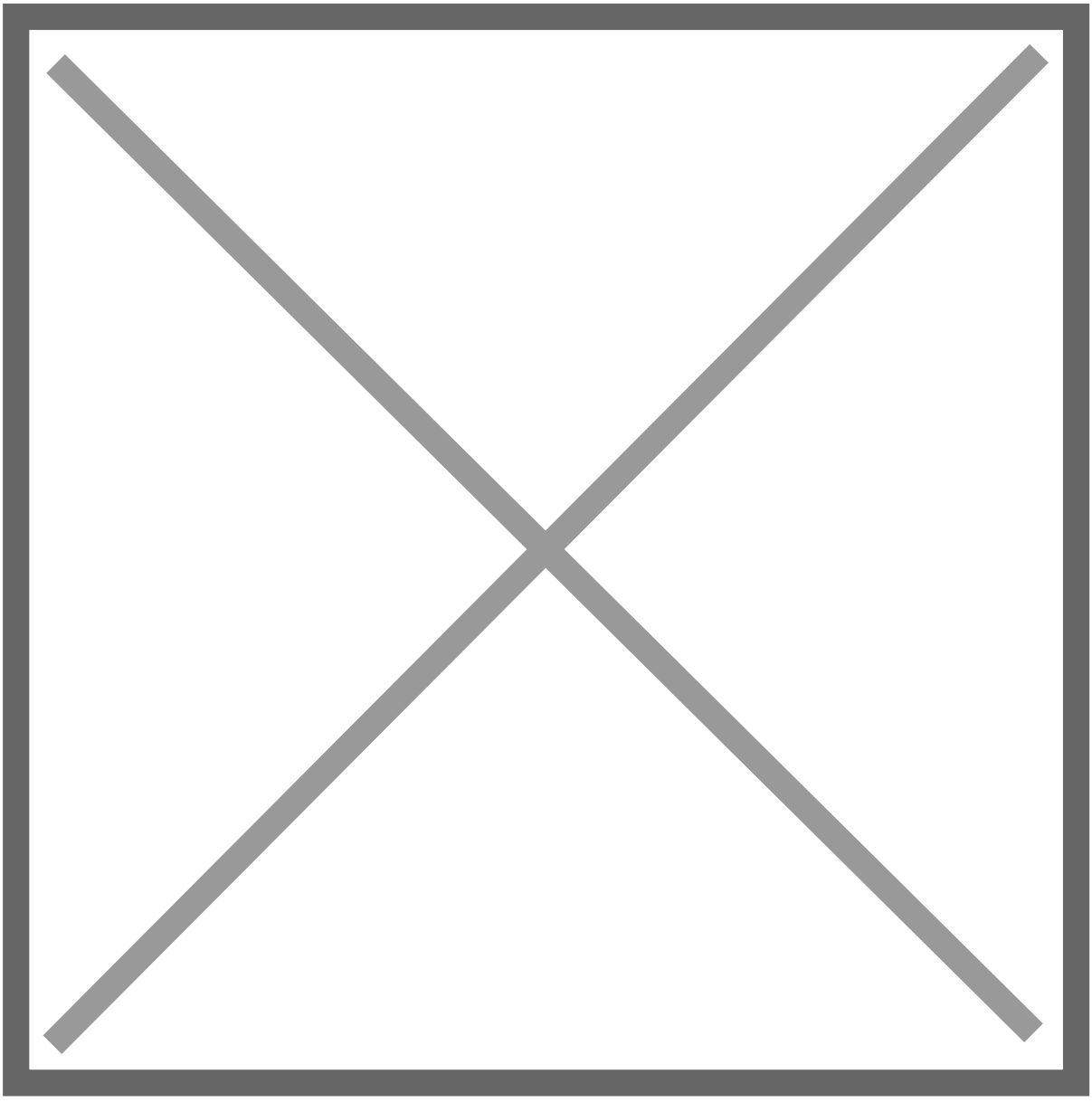
Синтаксис:

```
host [options] [hostname/IP]
```

Различные [options] управляют поведением команды, такими как тип запроса или стартовая запись (SOA) для указанного домена.

Пример:

```
host google.com
```



Вывод отображает разрешенные IPv4 и IPv6 адреса для предоставленного имени хоста.

hostname

Команда `hostname` предназначена для отображения и изменения имени хоста и домена системы, а также идентификации устройств в сетевой среде.

Воспользуйтесь командой для вывода, изменения или поиска имен хостов.

Синтаксис:

```
hostname [options] [name]
```

Параметр [options] определяет, что отображается в результате выполнения команды, в то время как параметр [name] временно устанавливает имя хоста в предоставленное имя.

Пример:

Для временного изменения имени системы выполните команду без параметров и укажите имя:

```
sudo hostname [name]
```

При выполнении команды не выводится результат. Для проверки текущего имени хоста выполните:

```
hostname
```

На экран выводится текущее имя хоста.

ping

Команда `ping` – это сетевая утилита для проверки доступности хоста. Она отправляет ICMP-запросы к хосту (компьютеру или серверу) и измеряет время, затраченное на возврат ответа (RTT).

Пинг помогает оценить сетевую задержку между двумя узлами и проверить, доступна ли сеть.

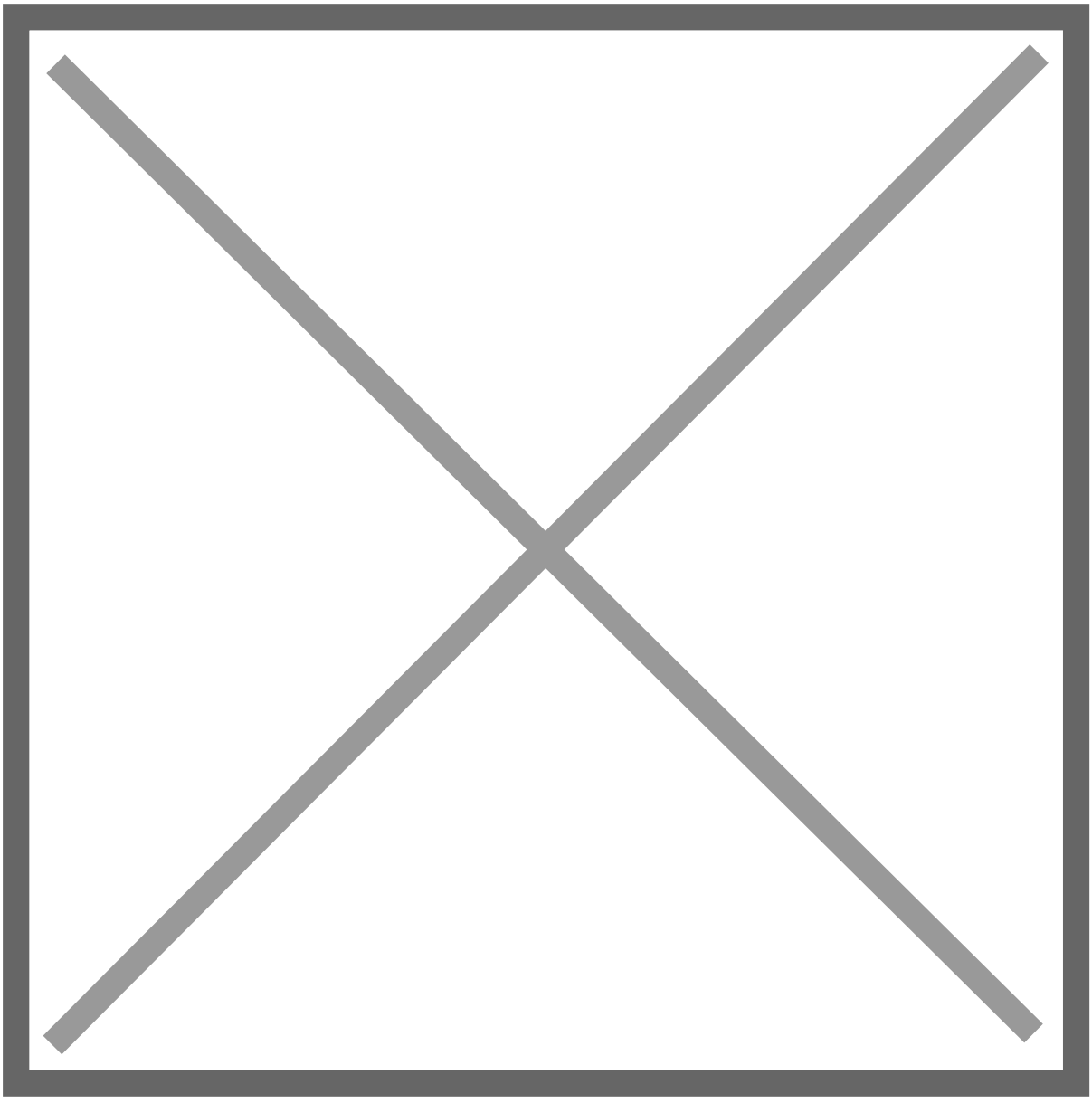
Синтаксис:

```
ping [options] [hostname/IP]
```

Укажите [hostname/IP] хоста, к которому выполняется пинг. Дополните команду опциями для управления её поведением, такими как количество запросов пинга, интервалы или размер пакета.

Пример:

```
ping -c 5 google.com
```



Эта команда посылает пять ICMP-пакетов на указанный хост и выводит статистику.

SS

Команда `ss` – это инструмент командной строки для отображения сетевой статистики. Этот инструмент включен в пакет `iproute2` и является более быстрой альтернативой команде `netstat`.

Для изучения сетевых сокетов и просмотра различных сетевых данных используйте команду `ss`.

Синтаксис:

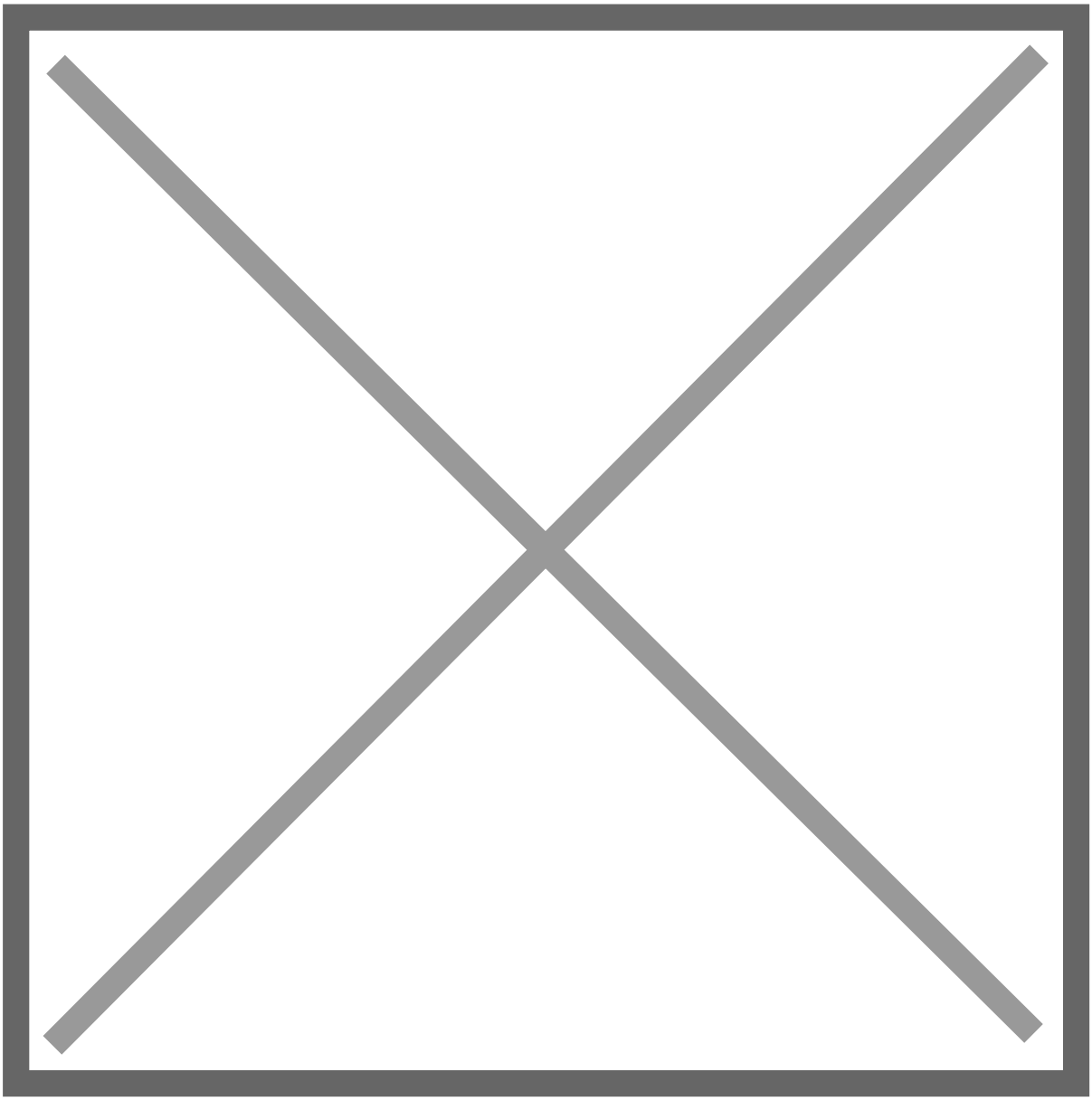
```
ss [options] [filter]
```

Параметр [options] дает возможность фильтровать сокеты по протоколу, в то время как параметр [filter] помогает упорядочивать сокеты по состоянию, чтобы сузить результаты.

Пример:

Для примера, чтобы показать все прослушивающие TCP-сокеты с помощью команды `ss`, введите опции -lt:

```
ss -lt
```



Результат показывает все TCP-сокеты в состоянии LISTEN, готовые к принятию входящих подключений.

route

Команда `route` в Linux предназначена для отображения и настройки таблицы маршрутизации. Она изменяет таблицы маршрутизации IP в ядре и помогает устанавливать статические маршруты к определенным хостам или сетям.

Примените эту команду после настройки сетевого интерфейса с помощью инструмента, например, команды `ifconfig`.

> Заметьте: более предпочтительной альтернативой команде `route` является команда `ip route`.
>
> *NETWORK ADMIN*

Синтаксис:

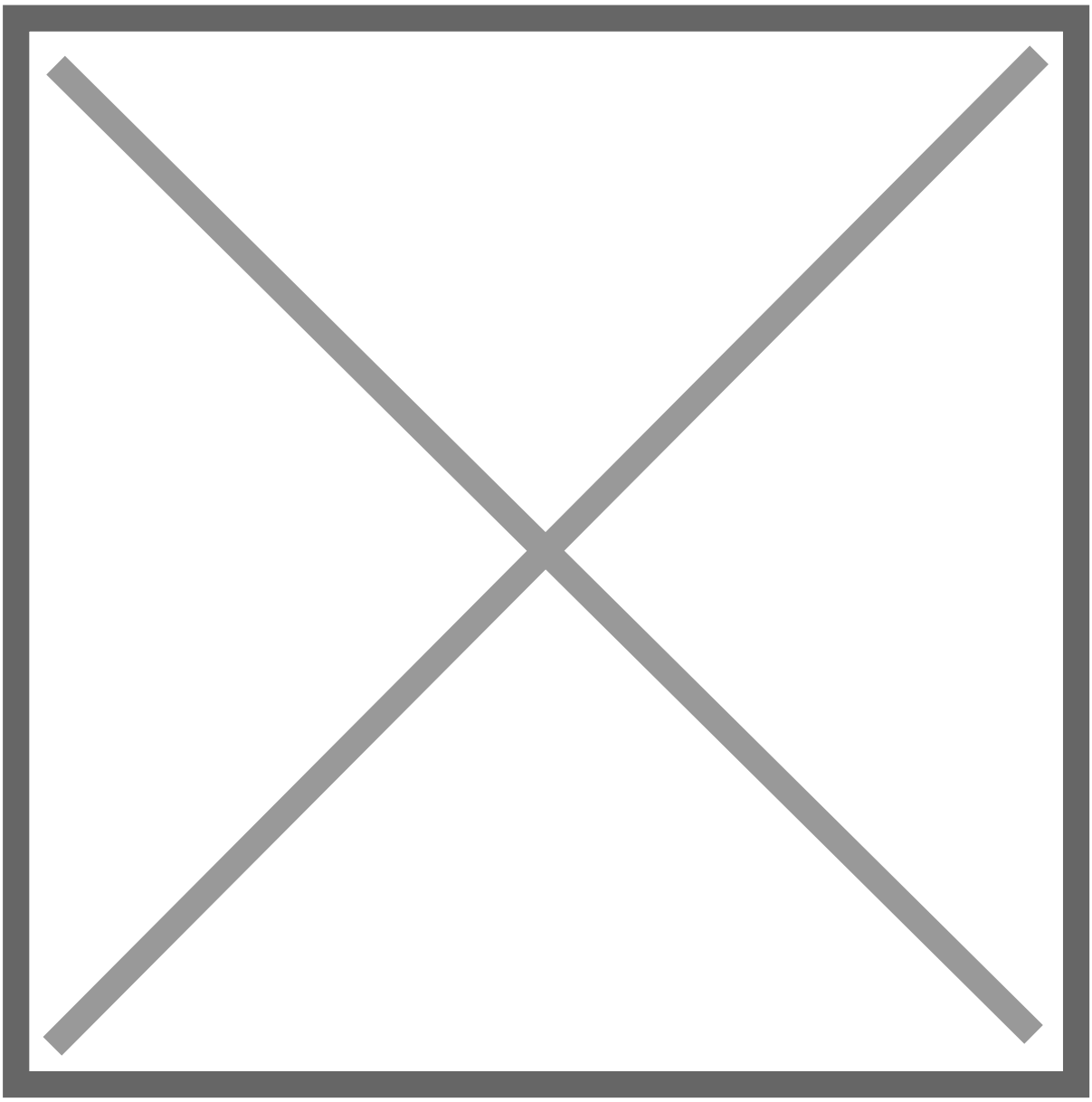
```
route [options] [subcommand] [arguments]
```

Включает следующие компоненты:

- **options** – Опциональные параметры командной строки, которые управляют видом вывода, семейством адресов и IP-протоколом.
- **subcommand** – Действие, которое нужно выполнить, такое как добавление или удаление.
- **arguments** – Дополнительные аргументы, которые различаются в зависимости от подкоманды.

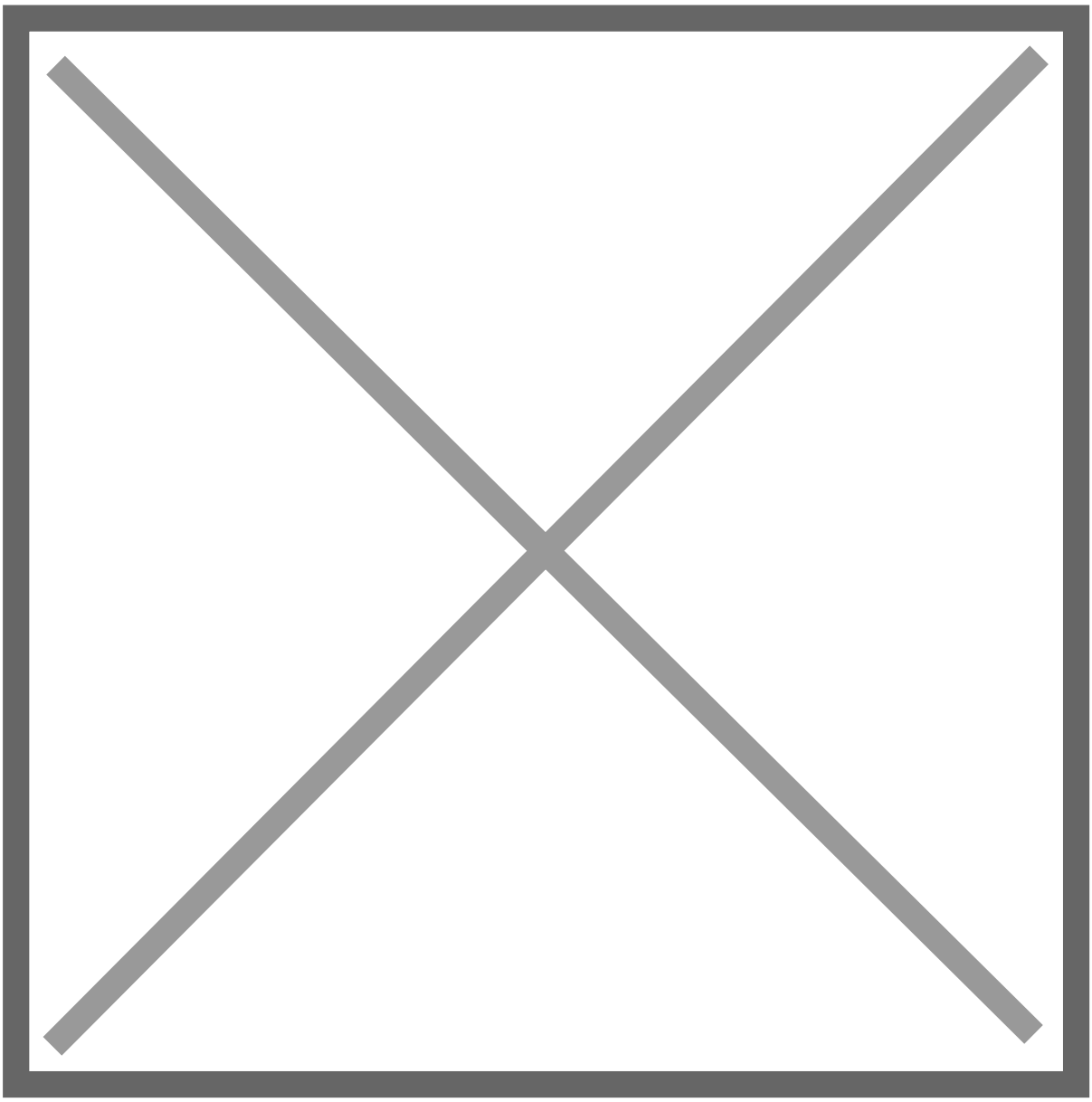
Пример:

```
route
```



Чтобы установить шлюз по умолчанию, используйте указанный формат:

```
sudo route add default gw [gateway]
```



Эта команда создает маршрут по умолчанию, который используется, если не существует других подходящих маршрутов. Указанный шлюз должен быть непосредственно доступным маршрутом.

arp

Команда `arp` показывает и настраивает кэш ARP (протокола разрешения адресов). Протокол ARP сопоставляет IP-адреса с физическими MAC-адресами в локальной сети. Кэш хранит такие соответствия для всех устройств в локальной сети.

Синтаксис:

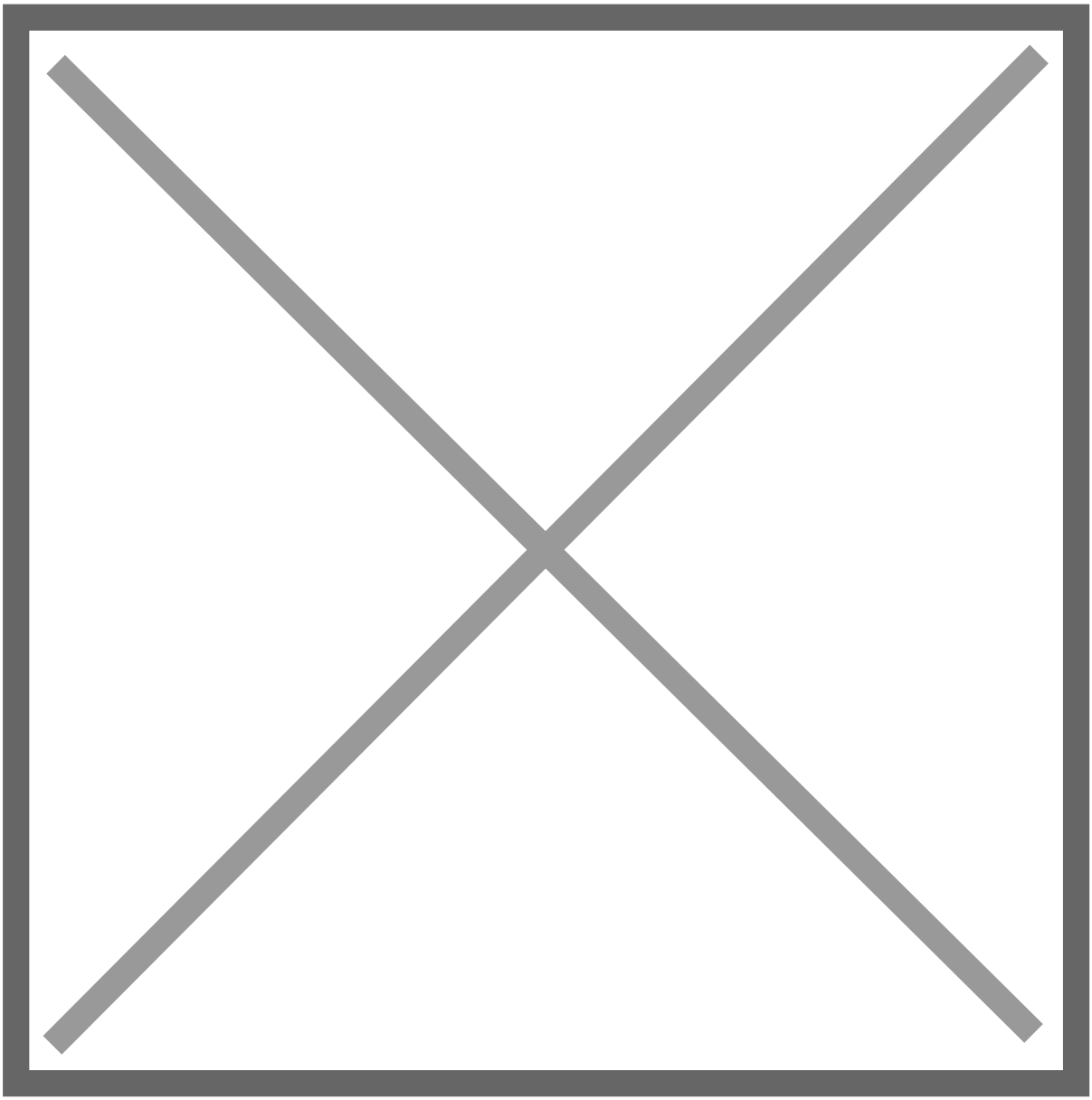
```
arp [options] [hostname/IP]
```

- Параметр [options] влияет на поведение команды, позволяя, например, настраивать и удалять действия или контролировать вывод.
- Параметр [hostname/IP] является необязательным идентификатором для удаленной системы, для которой нужно разрешить MAC-адрес. Если не указан, команда проверяет локальный кэш ARP.

Пример:

Чтобы вывести кэш ARP, выполните команду `arp` без дополнительных параметров:

```
arp
```



Вывод представляет собой таблицу с кэшем ARP (IP и MAC-адреса).

iwconfig

Команда `iwconfig` позволяет отображать и настраивать информацию о беспроводных сетевых интерфейсах и пригодится при решении проблем в беспроводных сетях.

Используйте эту команду для просмотра или изменения имени беспроводной сети, настроек управления питанием и других настроек беспроводной связи.

Синтаксис:

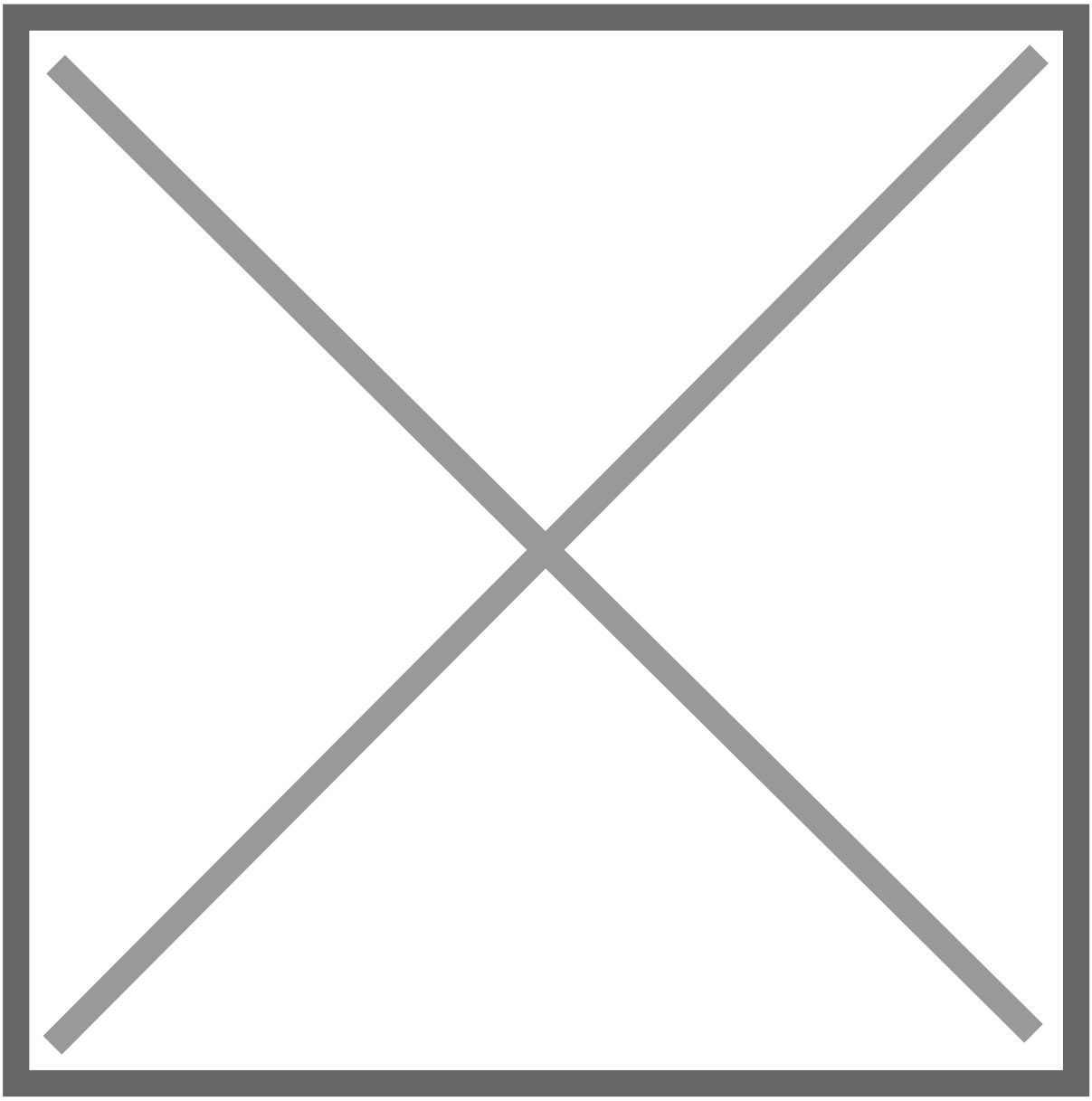
```
iwconfig [interface] [options]
```

Параметр [interface] позволяет фильтровать беспроводный сетевой интерфейс по имени, а параметр [options] управляет различными настройками, такими как режим работы, ограничение скорости и ключ шифрования беспроводной сети.

Пример:

Для просмотра доступных беспроводных интерфейсов на системе и их текущей настройки выполните команду без дополнительных параметров:

```
iwconfig
```



Эта команда выводит полную информацию о беспроводных интерфейсах в системе.

curl or wget

Команды `wget` и `curl` представляют собой инструменты командной строки для загрузки файлов из Интернета. Оба инструмента схожи, но имеют некоторые различия в способе работы и доступных опциях.

- Команда `wget` предназначена для загрузки файлов из Интернета с использованием протоколов HTTP, HTTPS или FTP. Этот инструмент легок в использовании для скачивания файлов.
- Команда `curl` универсальна и поддерживает различные сетевые протоколы, включая SCP, IMAP, POP3, SMTP и другие. Этот инструмент также отправляет HTTP-запросы и взаимодействует с веб-сервисами.

Для проверки скорости загрузки по сети используйте `curl` или `wget`.

Синтаксис:

```
wget [options] [URL]
```

```
curl [options] [URL]
```

Параметр [options] управляет различными параметрами загрузки и вывода, а параметр [URL] представляет собой URL-адрес файла для загрузки. Команда `curl` обладает более продвинутыми опциями и методами использования по сравнению с командой `wget`.

Пример:

Для скачивания файла с помощью команды `wget` используйте этот формат:

```
wget -O [file name] [URL]
```

Или же, чтобы сделать то же самое с помощью curl, выполните:

```
curl -o [file name] [URL]
```

Файл загружается с указанного URL и сохраняется с указанным именем файла.

mtr

Команда `mtr` (my traceroute) – это инструмент для диагностики, который объединяет функции команд `ping` и `traceroute`. Этот инструмент предоставляет актуальную информацию о качестве сети, что делает его отличным средством для устранения проблем с высокой задержкой и потерей пакетов.

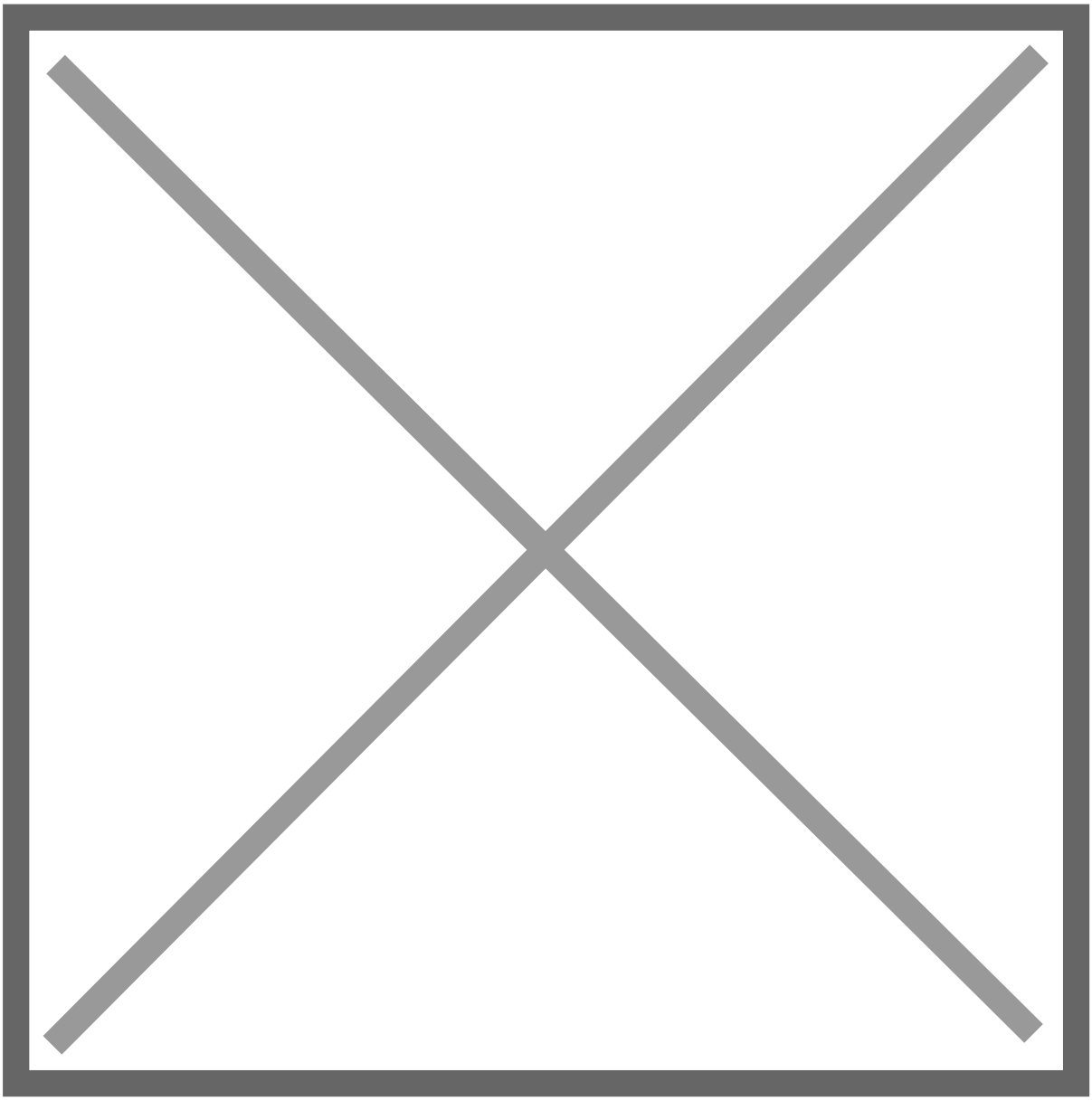
Синтаксис:

```
mtr [options] [hostname/IP]
```

Параметр [options] позволяет настраивать количество и размер пакетов, а параметр [hostname/IP] указывает на место назначения.

Пример:

```
mtr google.com
```



Для выхода из окна, нажмите “q”.

whois

Команда `whois` предназначена для запроса информации о доменных именах, IP-адресах и других сетевых данных. Используйте эту команду для получения данных о владельце домена, дате регистрации и дате истечения срока действия домена.

Синтаксис:

```
whois [options] [query]
```

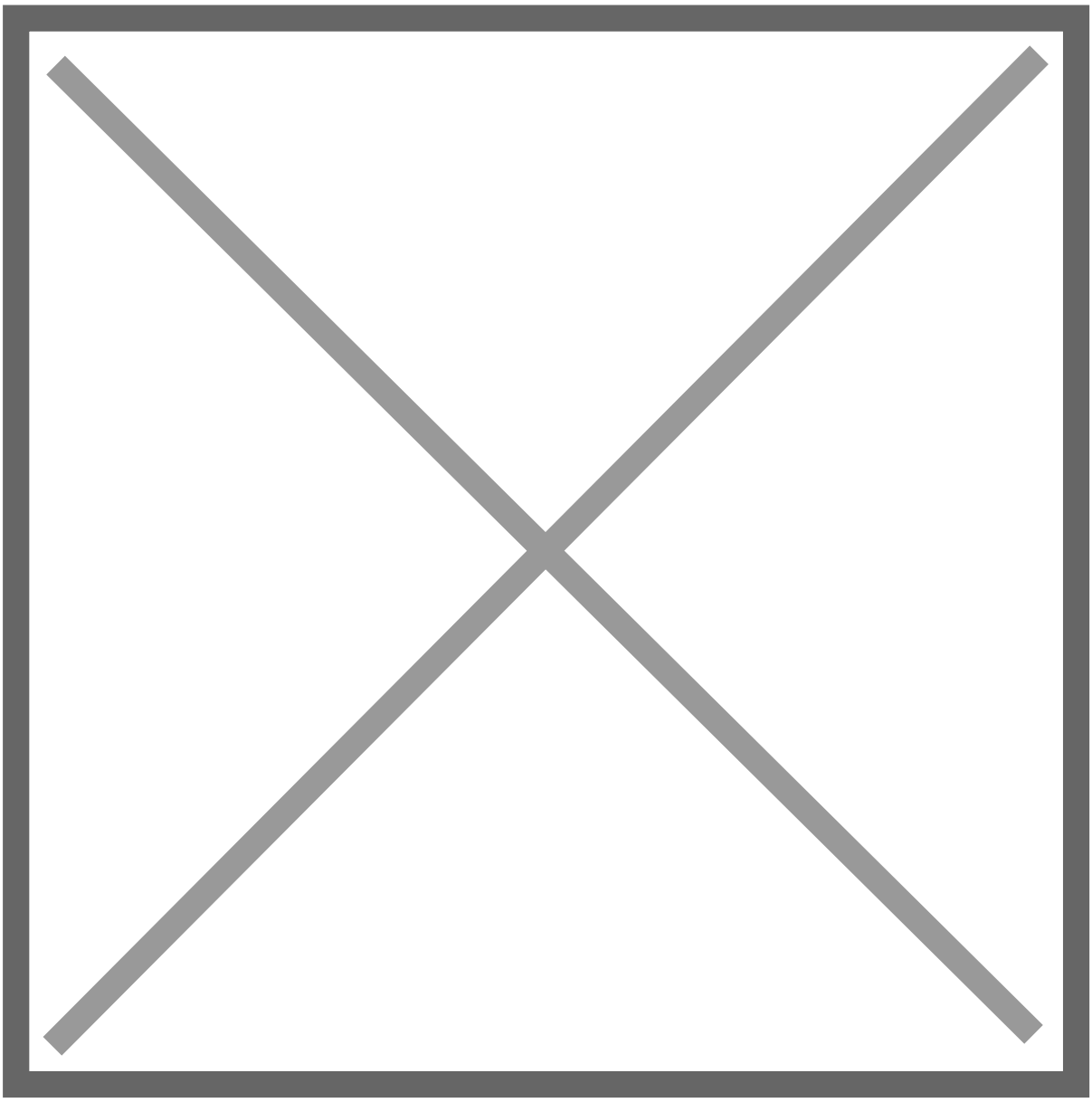
- Параметр [options] дает возможность указать конкретный сервер WHOIS для запроса, изменить протокол и добавить дополнительные параметры запроса.

- Параметр [query] может быть доменным именем, IP-адресом или номером автономной системы (ASN) для поиска информации.

Пример:

Чтобы выполнить простой запрос для указанного доменного имени, выполните команду без дополнительных опций. Например:

```
whois google.com
```



Вывод представляет результаты основного поиска WHOIS для указанного доменного имени.

iftop

Команда `iftop` – это утилита мониторинга сети. С ее помощью можно отслеживать сетевые соединения и использование пропускной способности в режиме реального времени.

Синтаксис:

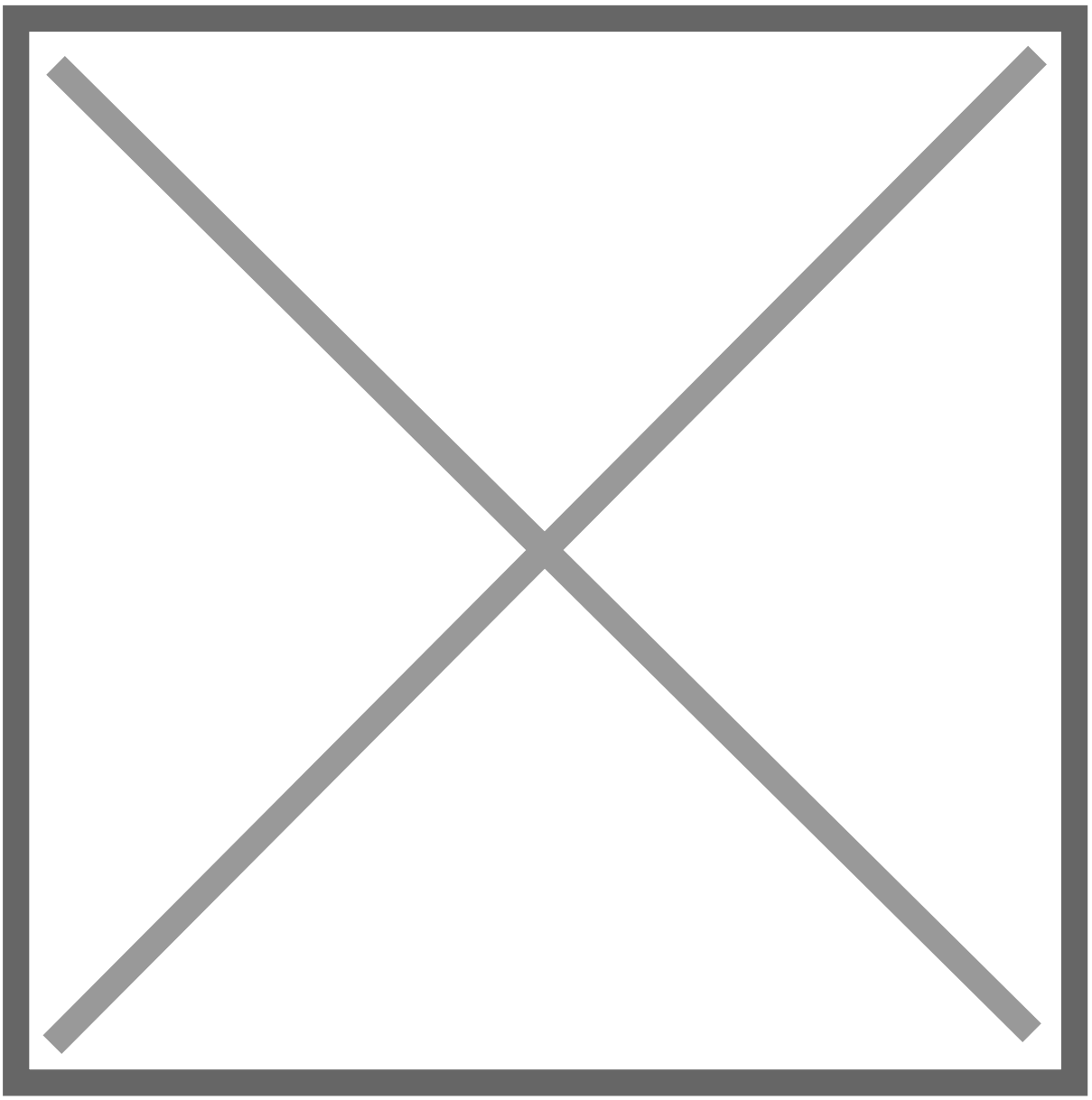
```
iftop [options]
```

Параметр [options] регулирует отображаемую информацию. Кроме того, для мониторинга всего трафика на сетевом интерфейсе необходимы соответствующие права доступа.

Пример:

Основное использование `iftop` не предполагает дополнительных опций:

```
sudo iftop
```



При выполнении команды открывается новый экран мониторинга, который обновляется при передаче данных через сетевой интерфейс.

С интерфейса можно управлять отображением на экране мониторинга, например, переключать вид на источник (s) или назначение (d). Для выхода из экрана, нажмите “q”.

tcpdump

Команда `tcpdump` представляет собой инструмент для сниффинга пакетов и обеспечения сетевой безопасности, который захватывает информацию о сетевых пакетах в реальном времени. Используйте эту команду для анализа трафика, устранения проблем и мониторинга сетевой безопасности.

Синтаксис:

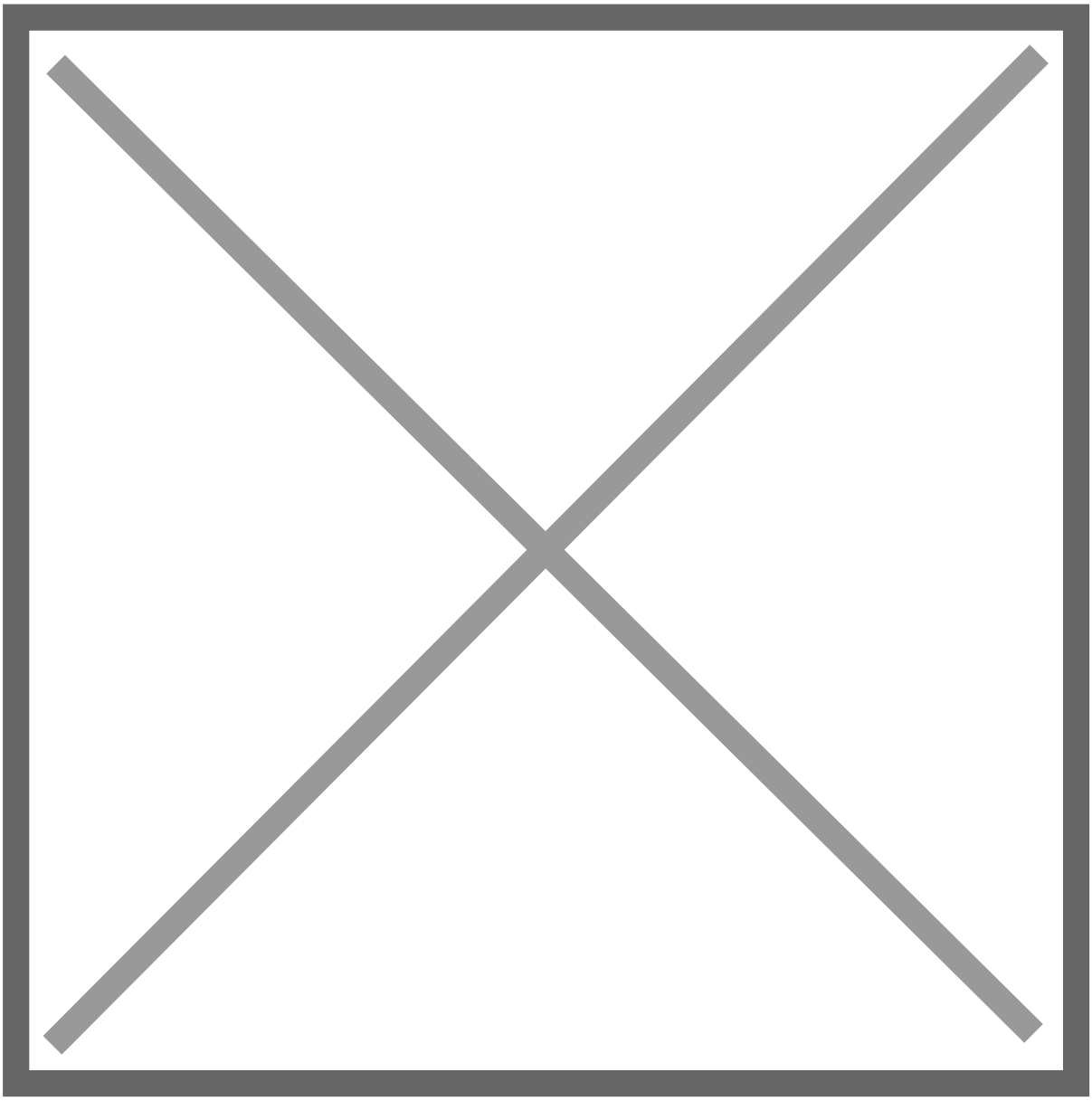
```
tcpdump [options] [filter]
```

Параметр [options] управляет различными параметрами отображения, контролирует количество захваченных пакетов и позволяет работать с файлами. Для установки критериев захвата пакетов используйте параметр [filter].

Пример:

Для захвата пакетов на конкретном порту используйте следующий формат:

```
sudo tcpdump port 80
```



Фильтр `port 80` используется для захвата пакетов на определенном порту с целью мониторинга HTTP-трафика.

ifplugstatus

Команда `ifplugstatus` – это простая утилита для проверки статуса сетевого интерфейса. Она помогает определить, подключен ли сетевой кабель к интерфейсу Ethernet.

Для проверки физической связи в сети, особенно после внесения изменений в сетевой интерфейс, используйте `ifplugstatus`.

Синтаксис:

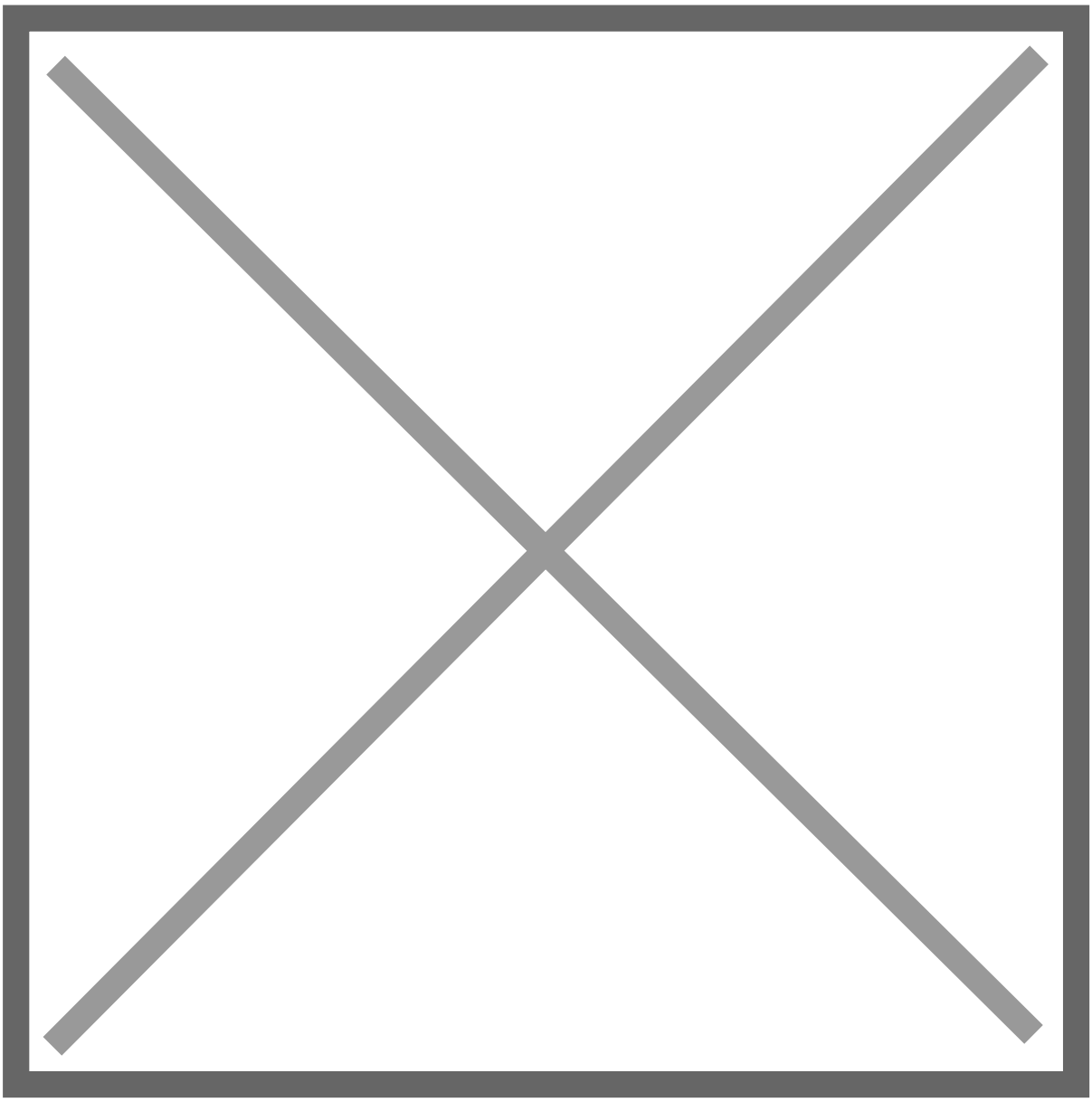
```
ifplugstatus [options] [interface]
```

Параметр [options] позволяет установить конкретный файл конфигурации или запустить команду в пакетном режиме для скриптов. Укажите параметр [interface], чтобы проверить статус указанного сетевого интерфейса.

Пример:

Для отображения статуса всех сетевых интерфейсов выполните команду без дополнительных параметров:

```
ifplugstatus
```



Если в выводе указано “link beat detected”, это свидетельствует о наличии активной физической связи у интерфейса.

Заключение

После прочтения этого руководства вы освоите основные команды для настройки сети в Linux. Обращайтесь к этой статье, когда вам понадобится определенная команда.

Revision #1
Created 2023-11-03 19:28:17 UTC by odiljonov
Updated 2023-11-03 19:28:52 UTC by odiljonov