

ГУАП

КАФЕДРА № 44

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

ассистент

должность, уч. степень, звание

подпись, дата

К.А.Белов

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

по курсу: АДМИНИСТРИРОВАНИЕ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ НА БАЗЕ UNIX

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

подпись, дата

Е.Д.Тегай

инициалы, фамилия

Санкт-Петербург 2025

ЧАСТЬ 1. ЗАГРУЗКИ ОС. УПРАВЛЕНИЕ СЕТЬЮ. УПРАВЛЕНИЕ СЕРВИСАМИ.

Цель работы

Приобретение навыков по управлению процессом загрузки ОС и управлению службами. Изучение основных команд управления режимами работы и службами. Приобретение навыков управления сетевыми интерфейсами и сетевой службой. Изучение команд смены пользователя.

Ход работы

Первым делом необходимо загрузить систему в графический режим и запустить виртуальную машину. Сразу отметим, что в качестве ОС была выбрана Astra Linux Common Edition. Результатом данных действий является скриншот на рисунке 1, который показывает их корректность.

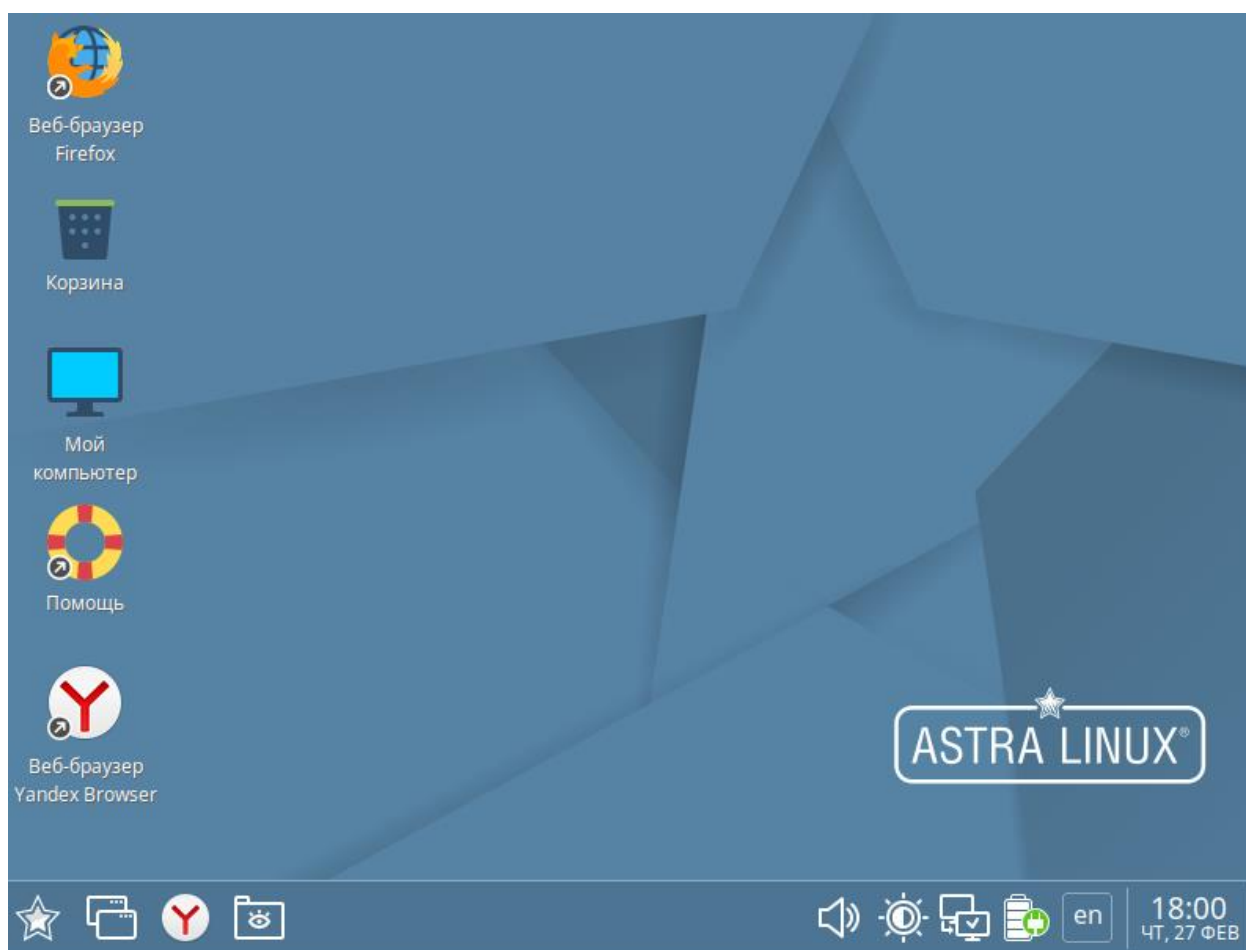


Рисунок 1 – Главное окно после запуска и входа

Затем необходимо определить текущий режим работы системы. Используемая для этого команда и результат продемонстрированы на рисунке

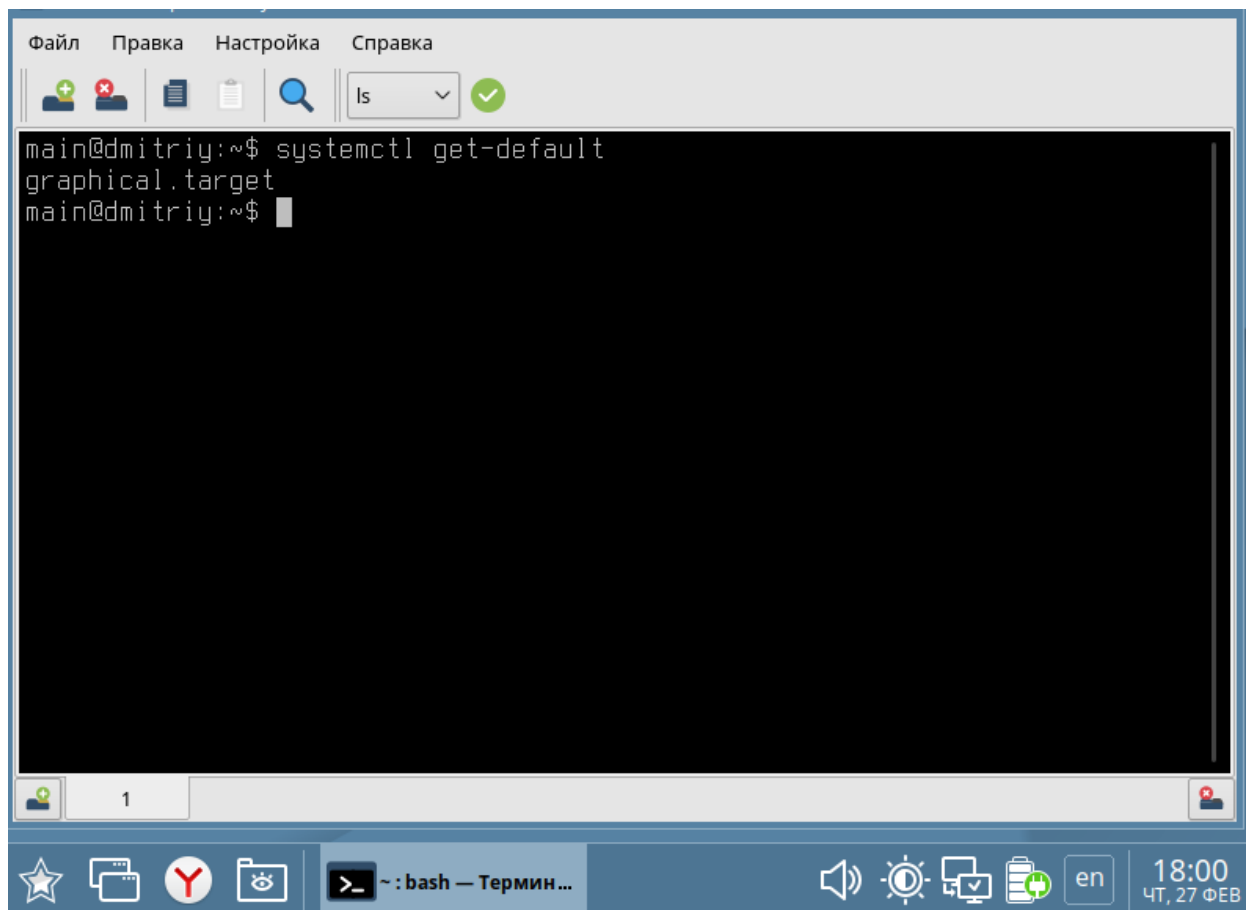


Рисунок 2 – Определение текущего режима работы системы

Исходя из вывода на рисунке 2 можно понять, что система загружена в графическом режиме. Затем необходимо определить список установленных сервисов. Используемая для этого команда и вывод показаны на рисунке 3.

```
main@dmitriy:~$ systemctl list-units --type=service
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
acpid.service                      loaded active running ACPI event daemon
alsa-restore.service               loaded active exited Save/Restore Sound Card S
astra-orientation.service          loaded active running Screens orientation monit
avahi-daemon.service               loaded active running Avahi mDNS/DNS-SD Stack
console-setup.service              loaded active exited Set console font and keym
cron.service                       loaded active running Regular background progra
cups.service                       loaded active running CUPS Scheduler
dbus.service                       loaded active running D-Bus System Message Bus
fly-dm.service                    loaded active running The FLY login manager
getty@tty1.service                 loaded active running Getty on tty1
gpm.service                        loaded active running LSB: gpm sysv init script
keyboard-setup.service             loaded active exited Set the console keyboard
kmod-static-nodes.service          loaded active exited Create list of required s
libflygetexe-bin.service           loaded active running The FLY get exec service
networking.service                loaded active exited Raise network interfaces
NetworkManager.service            loaded active running Network Manager
oFono.service                      loaded active running oFono Mobile telephony st
polkit.service                     loaded active running Authorization Manager
```

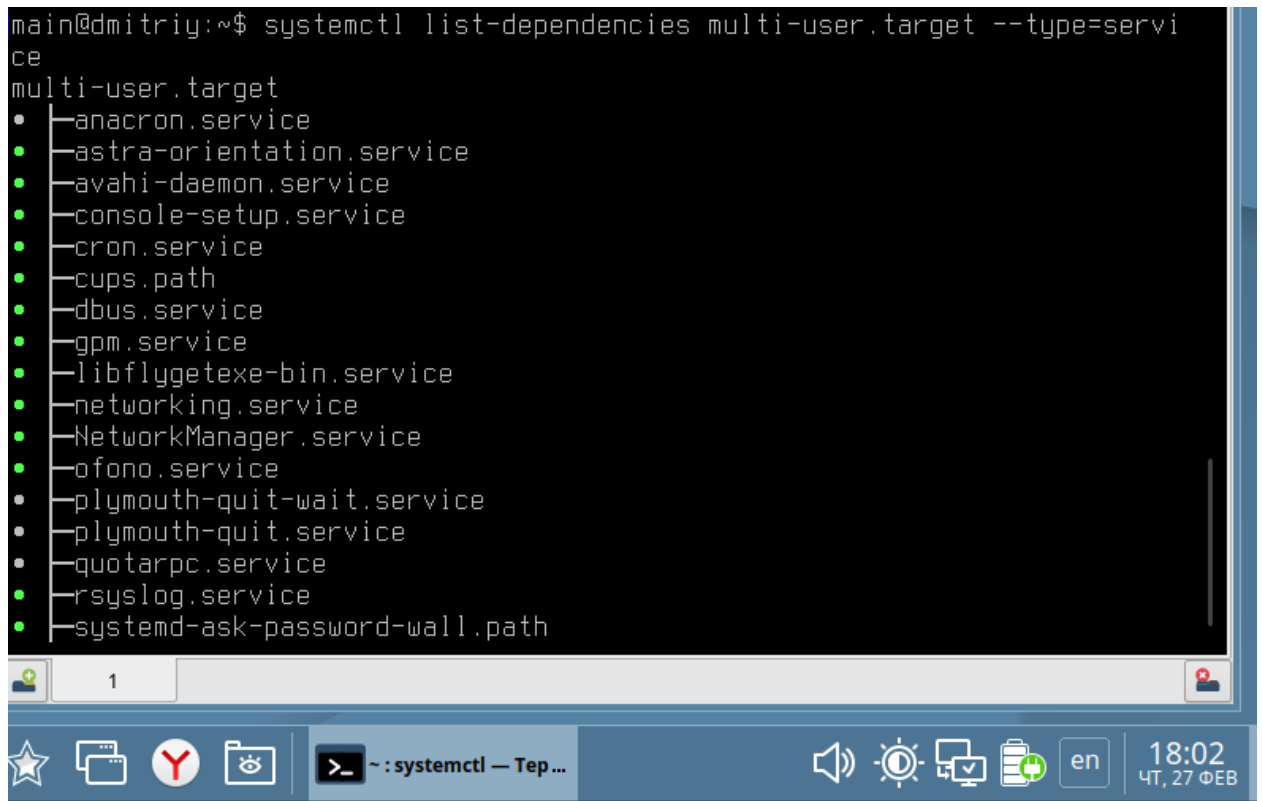
Рисунок 3 – Определение списка установленных сервисов

В случае, как на рисунке 3, показывается список запущенных (и разрешённых) сервисов и их состояние. Для просмотра всех (не только разрешённых) сервисов используется команда, изображённая на рисунке 4.

```
main@dmitriy:~$ systemctl list-unit-files --type=service
UNIT FILE                           STATE
acpi-support.service                disabled
acpid.service                       disabled
alsa-restore.service                static
alsa-state.service                  static
alsa-utils.service                  masked
anacron.service                     enabled
apt-daily-upgrade.service            static
apt-daily.service                   static
astra-orientation.service            enabled
autovt@.service                     enabled
avahi-daemon.service                enabled
bootlogd.service                    masked
bootlogs.service                    masked
bootmisc.service                    masked
checkfs.service                     masked
checkroot-bootclean.service          masked
checkroot.service                   masked
connman-wait-online.service           enabled
```

Рисунок 4 – Определение списка установленных сервисов

Следующим действием является определение списка сервисов, запускаемых в 3-ем режиме работы (multi-user.target, многопользовательский режим без графического интерфейса). Для этого используется команда, изображённая на рисунке 5. Она показывает все сервисы, которые загружаются в этом режиме.



```
main@dmritriy:~$ systemctl list-dependencies multi-user.target --type=service
multi-user.target
• └─anacron.service
• └─astra-orientation.service
• └─avahi-daemon.service
• └─console-setup.service
• └─cron.service
• └─cups.path
• └─dbus.service
• └─gpm.service
• └─libflygetexe-bin.service
• └─networking.service
• └─NetworkManager.service
• └─ofono.service
• └─plymouth-quit-wait.service
• └─plymouth-quit.service
• └─quotarpc.service
• └─rsyslog.service
• └─systemd-ask-password-wall.path
```

Рисунок 5 – Определение списка сервисов, запускаемых в 3-ем режиме работы

Далее необходимо определить состояние сетевого сервиса. Для этого используется команда, изображённая на рисунке 6. Она показывает, запущен ли сервис, его текущее состояние и последние логи.

```
• |plymouth-quit.service
• |quotarpc.service
• |rsyslog.service
• |systemd-ask-password-wall.path
• |systemd-logind.service

[3]+ Остановлен      systemctl list-dependencies multi-user.target --type=s
ervice
main@dmritriy:~$
main@dmritriy:~$ systemctl status NetworkManager.service
• NetworkManager.service - Network Manager
  Loaded: loaded (/lib/systemd/system/NetworkManager.service; enabled; ve
  Active: active (running) since Thu 2025-02-27 17:44:34 MSK; 18min ago
    Docs: man:NetworkManager(8)
  Main PID: 459 (NetworkManager)
    Tasks: 5 (limit: 4915)
   CGroup: /system.slice/NetworkManager.service
           └─459 /usr/sbin/NetworkManager --no-daemon
              625 /sbin/dhclient -d -q -sf /usr/lib/NetworkManager/nm-dhcp-
lines 1-9/9 (END)
```

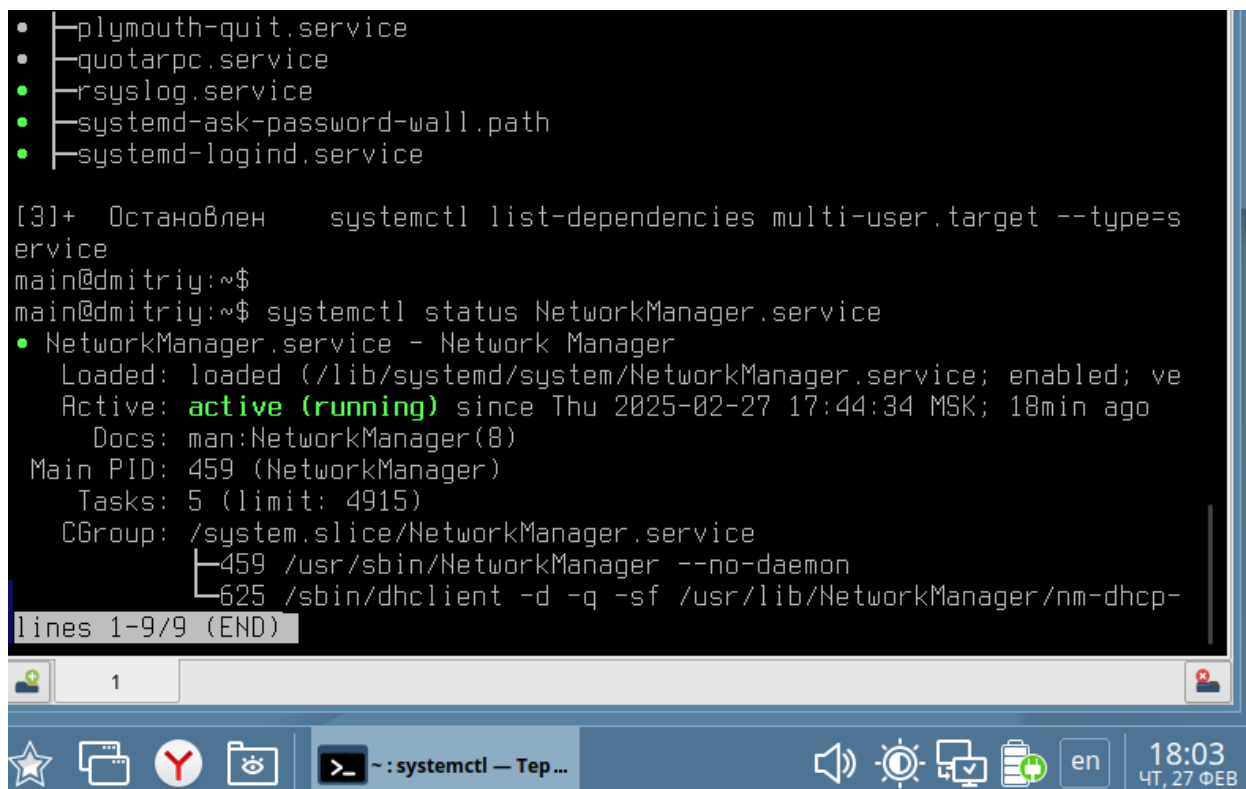


Рисунок 6 – Определение состояния сетевого сервиса

Далее необходимо определить параметры сетевых интерфейсов. Так как на данной машине установлен пакет `net-tools` (в чём можно удостовериться, набрав команду `dpkg -l | grep net-tools`), можно воспользоваться командой, изображённой на рисунке 7 (просто написать команду `ifconfig` – а не удалось, потому что исполняемые файлы этого пакета находятся не в `$PATH`).

```

main@dmitriy:~$ /sbin/ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.3.15 netmask 255.255.255.0 broadcast 10.0.3.255
    inet6 fe80::3a3:e774:cc0b:db52 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:9a:da:cd txqueuelen 1000 (Ethernet)
    RX packets 28 bytes 4458 (4.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 51 bytes 5449 (5.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 28 bytes 1652 (1.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 28 bytes 1652 (1.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

main@dmitriy:~$

```

Рисунок 7 – Определение параметров сетевых интерфейсов

Но можно также воспользоваться и командой, показанной на рисунке 8.

```

main@dmitriy:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP group default qlen 1000
    link/ether 08:00:27:9a:da:cd brd ff:ff:ff:ff:ff:ff
    inet 10.0.3.15/24 brd 10.0.3.255 scope global noprefixroute dynamic et
h0
        valid_lft 85080sec preferred_lft 85080sec
    inet6 fe80::3a3:e774:cc0b:db52/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
main@dmitriy:~$

```

Рисунок 8 – Определение параметров сетевых интерфейсов

Далее необходимо определить статические маршруты сети. Это делается с помощью команды, изображённой на рисунке 9.

```
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP group default qlen 1000
link/ether 08:00:27:9a:da:cd brd ff:ff:ff:ff:ff:ff
inet 10.0.3.15/24 brd 10.0.3.255 scope global noprefixroute dynamic et
h0
valid_lft 85080sec preferred_lft 85080sec
inet6 fe80::3a3:e774:cc0b:db52/64 scope link noprefixroute
valid_lft forever preferred_lft forever
main@dmitriy:~$ /sbin/route -n
Kernel IP routing table
Destination      Gateway         Genmask        Flags Metric Ref    Use If
ace
0.0.0.0          10.0.3.2       0.0.0.0        UG    100    0      0 et
h0
10.0.3.0         0.0.0.0        255.255.255.0  U    100    0      0 et
h0
main@dmitriy:~$
```

Рисунок 9 – Определение статических маршрутов сети

Можно также воспользоваться командой, изображённой на рисунке 10.

```
main@katya:~$ ip route show
10.0.3.0/24 dev eth0 proto kernel scope link src 10.0.3.15 metric 100
main@katya:~$
```

Рисунок 10 – Определение статических маршрутов сети

Затем необходимо удалить статический маршрут по умолчанию. Это делается с помощью команды, показанной на рисунке 11. Убедиться в этом можно через повторное использование просмотра статических маршрутов сети, как это показано на рисунке 12.

```
main@katya:~$ sudo ip route del default
```

Рисунок 11 – Удаление дефолтного статического маршрута


```
main@katya:~$ /sbin/route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use If
ace
10.0.3.0         0.0.0.0        255.255.255.0  U        100    0      0 et
h0
main@katya:~$
```

Рисунок 12 – Результат удаления

Далее необходимо остановить сетевой сервис и после этого определить состояние сетевого сервиса. Это делается с помощью команд, показанных на рисунках 13-14. На рисунке 15 повторно используется команда для определения параметров сетевых протоколов. Это действие необходимо для убеждения в проделанных действиях. Однако, из-за того, что, в данном случае, настройки сети загружаются статически, вывод команды не меняется. Но можно убедиться в корректности действий по иконке внизу.

```
main@katya:~$ sudo systemctl stop NetworkManager
main@katya:~$
```

Рисунок 13 – Остановка всех сетевых сервисов

```
main@katya:~$ sudo systemctl status NetworkManager
• NetworkManager.service - Network Manager
  Loaded: loaded (/lib/systemd/system/NetworkManager.service; enabled; ve
  Active: inactive (dead) since Thu 2025-02-27 18:19:51 MSK; 4min 47s ago
  Docs: man:NetworkManager(8)
  Main PID: 479 (code=exited, status=0/SUCCESS)
  Tasks: 2 (limit: 4915)
  CGroup: /system.slice/NetworkManager.service
          └─635 /sbin/dhclient -d -q -sf /usr/lib/NetworkManager/nm-dhcp-

feb 27 18:11:56 katya NetworkManager[479]: <info> [1740669116.2744] manag
feb 27 18:11:56 katya NetworkManager[479]: <info> [1740669116.2759] polic
feb 27 18:11:56 katya NetworkManager[479]: <info> [1740669116.2764] devic
feb 27 18:11:56 katya NetworkManager[479]: <info> [1740669116.2876] manag
feb 27 18:11:56 katya NetworkManager[479]: <info> [1740669116.2936] manag
feb 27 18:12:15 katya NetworkManager[479]: <info> [1740669135.5112] manag
feb 27 18:19:51 katya systemd[1]: Stopping Network Manager...
feb 27 18:19:51 katya NetworkManager[479]: <info> [1740669591.1779] caught
feb 27 18:19:51 katya NetworkManager[479]: <info> [1740669591.1930] exiting
feb 27 18:19:51 katya systemd[1]: Stopped Network Manager.
```

Рисунок 14 – Статус сетевого сервиса

```
main@katya:~$ /sbin/ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.3.15 netmask 255.255.255.0 broadcast 10.0.3.255
    inet6 fe80::3a3:e774:cc0b:db52 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:9a:da:cd txqueuelen 1000 (Ethernet)
    RX packets 38 bytes 5690 (5.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 61 bytes 5788 (5.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 28 bytes 1676 (1.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 28 bytes 1676 (1.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

main@katya:~$
```

Рисунок 15 – Результат остановки (пропало соединение)

Далее необходимо с помощью команды **ifconfig** активизировать сетевой интерфейс с параметрами, определёнными выше. Это продемонстрировано на рисунке 16.

```
main@katya:~$ sudo ifconfig eth0 up
main@katya:~$
```

Рисунок 16 – Активирование сетевого интерфейса

После этого была повторно написана команда определения параметров сетевых интерфейсов. Это показано на рисунке 17.

```
main@katya:~$ /sbin/ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.3.15 netmask 255.255.255.0 broadcast 10.0.3.255
    inet6 fe80::3a3:e774:cc0b:db52 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:9a:da:cd txqueuelen 1000 (Ethernet)
    RX packets 38 bytes 5690 (5.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 61 bytes 5788 (5.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 28 bytes 1676 (1.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 28 bytes 1676 (1.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

main@katya:~$
```

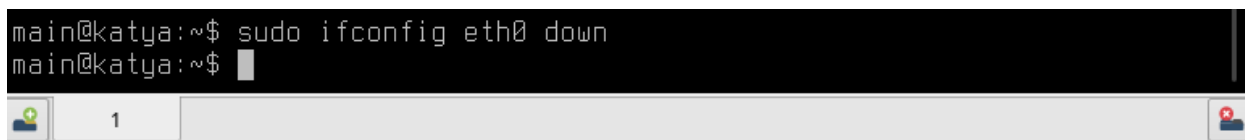
Рисунок 17 – Определение параметров сетевых интерфейсов

Затем снова необходимо определить статические маршруты сети. Это показано на рисунке 18.

```
main@katya:~$ sudo route
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use If
ace
10.0.3.0         0.0.0.0         255.255.255.0   U        100    0      0 et
h0
main@katya:~$
```

Рисунок 18 – Определение статических маршрутов сети

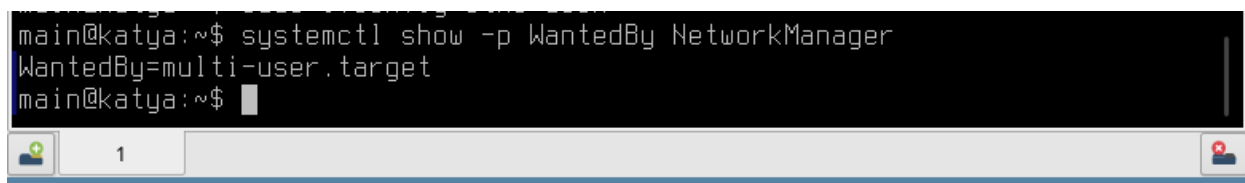
Затем необходимо деактивизировать сетевой интерфейс с помощью команды **ifconfig**. Это показано на рисунке 19.



```
main@katya:~$ sudo ifconfig eth0 down
main@katya:~$
```

Рисунок 19 – Деактивация сетевого интерфейса

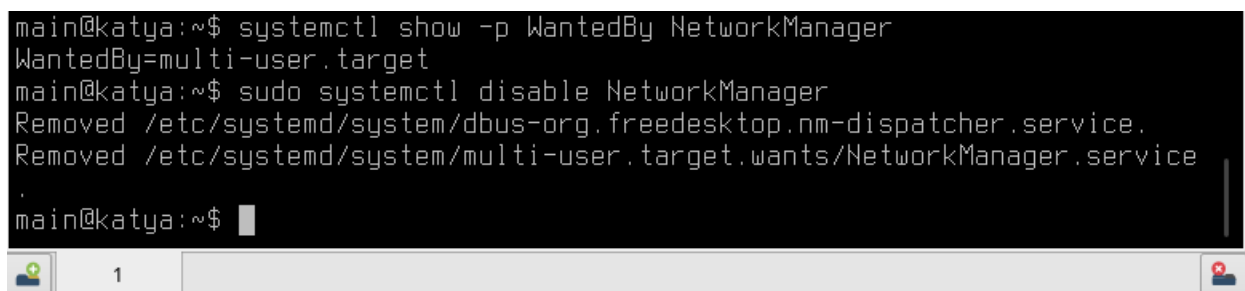
Далее нужно запустить сетевой сервис и получить его состояние во всех режимах работы. Это делается с помощью команд, изображённых на рисунке



```
main@katya:~$ systemctl show -p WantedBy NetworkManager
WantedBy=multi-user.target
main@katya:~$
```

Рисунок 20 – Запуск и просмотр состояния сетевого сервиса

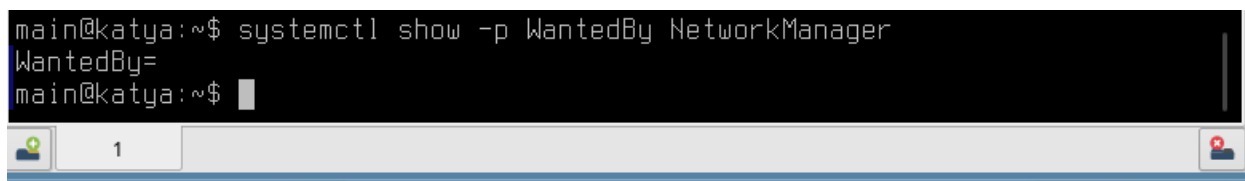
После этого необходимо отключить сетевой сервис в 3-ем режиме работы. Это делается с помощью команды, показанной на рисунке 21.



```
main@katya:~$ systemctl show -p WantedBy NetworkManager
WantedBy=multi-user.target
main@katya:~$ sudo systemctl disable NetworkManager
Removed /etc/systemd/system/dbus-org.freedesktop.nm-dispatcher.service.
Removed /etc/systemd/system/multi-user.target.wants/NetworkManager.service
main@katya:~$
```

Рисунок 21 – Отключение сетевого сервиса в 3-ем режиме работы

Далее необходимо получить состояние сетевого сервиса во всех режимах работы. Это показано на рисунке 22.



```
main@katya:~$ systemctl show -p WantedBy NetworkManager
WantedBy=
main@katya:~$
```

Рисунок 22 – Получение состояния сетевого сервиса во всех режимах работы

Затем необходимо перевести систему в 3-ий режим работы. Это делается с помощью команды:

sudo systemctl isolate multi-user.target

Результат работы команды показан на рисунке 23.

```
Astra Linux CE 2.12.46 (orel) katya tty1
katya login: root
root
Password:
root@katya:~#
```

Рисунок 23 – Переход в 3-ий режим работы

Здесь необходимо войти в систему под пользователем root и определить текущий режим работы системы. Все проделанные действия продемонстрированы на рисунке 24.

```
Astra Linux CE 2.12.46 (orel) katya tty1
katya login: root
root
Password:
root@katya:~# runlevel
5 3
root@katya:~#
```

Рисунок 24 – Определение режима работы системы

Далее снова необходимо определить параметры сетевых интерфейсов. Результат продемонстрирован на рисунке 25.

```
root@katya:~# ifconfig -a
eth0: flags=4098<BROADCAST,MULTICAST> mtu 1500
    inet 10.0.3.15 netmask 255.255.255.0 broadcast 10.0.3.255
    ether 08:00:27:9a:da:cd txqueuelen 1000 (Ethernet)
    RX packets 38 bytes 5690 (5.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 61 bytes 5788 (5.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 28 bytes 1676 (1.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 28 bytes 1676 (1.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@katya:~#
```

Рисунок 25 – Определение параметров сетевых интерфейсов

После этого необходимо определить статические маршруты сети. Это сделано с помощью команды, показанной на рисунке 26.

```
root@katya:~# route
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
root@katya:~#
```

Рисунок 26 – Определение статических маршрутов сети

Далее необходимо вернуть систему в 5-ый режим работы. Это делается с помощью команды, показанной на рисунке 27.

```
root@katya:~# sudo systemctl isolate graphical.target_
```

Рисунок 27 – Возвращение в 5-ый режим работы

Результат работы команды показан на рисунке 28.

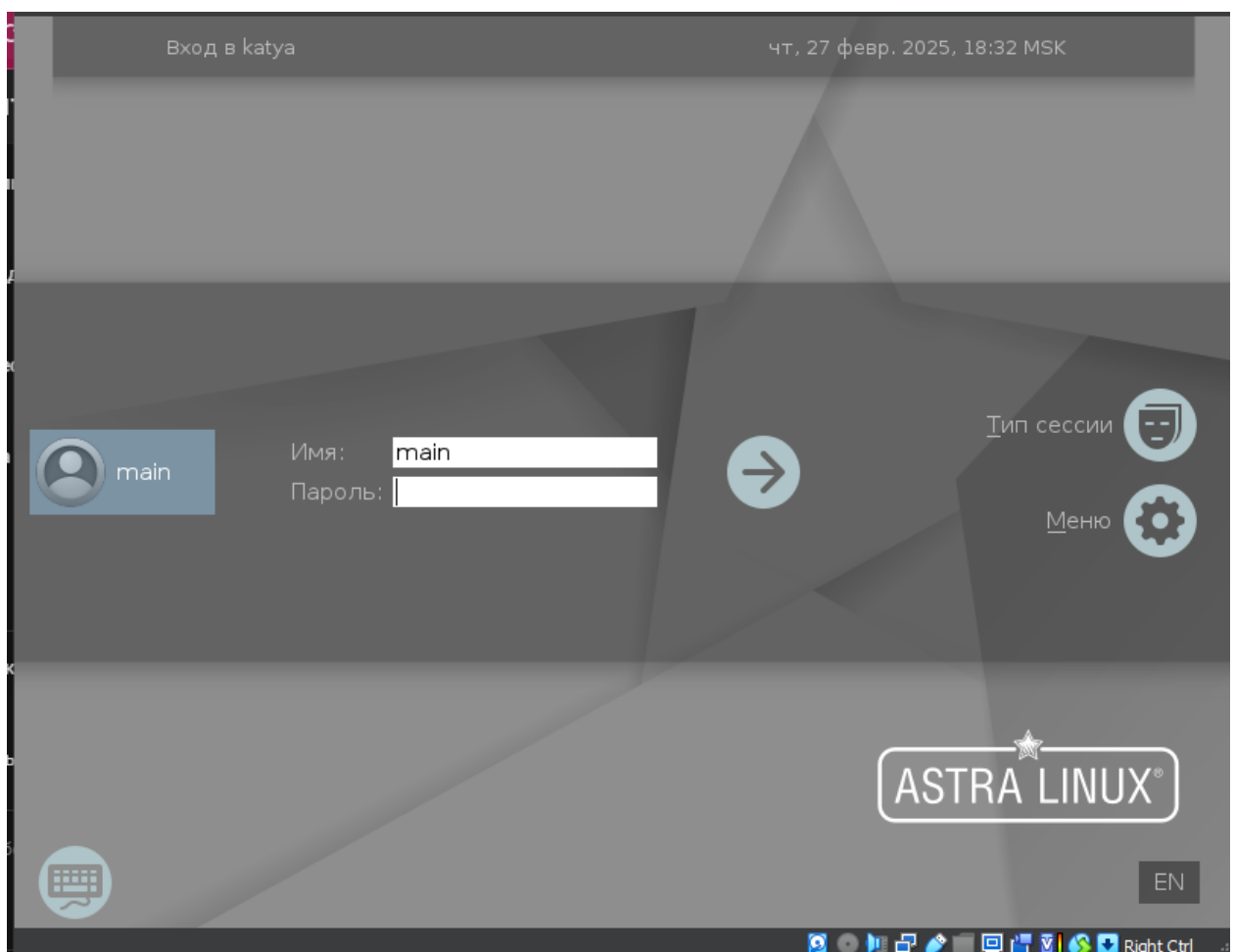


Рисунок 28 – Результат работы команды

Снова определим текущий режим работы. Это показано на рисунке 29.

```
main@katya:~$ su -
Пароль:
root@katya:~# runlevel
3 5
root@katya:~# █
```

Рисунок 29 – Определение текущего режима работы

Далее необходимо включить сетевой сервис в 3 режиме работы. Для этого используются команды, показанные на рисунках 30.

```
root@katya:~# sudo systemctl enable NetworkManager
Created symlink /etc/systemd/system/multi-user.target.wants/NetworkManager.service → /lib/systemd/system/NetworkManager.service.
Created symlink /etc/systemd/system/dbus-org.freedesktop.nm-dispatcher.service → /lib/systemd/system/NetworkManager-dispatcher.service.
root@katya:~# █
```

Рисунок 30 – Обеспечение включения автоматического запуска

```
main@katya:~$ sudo systemctl isolate multi-user.target █
```

Рисунок 31 – Переход в 3 режим

```
root@katya:~# sudo systemctl start NetworkManager
sudo systemctl start NetworkManager
root@katya:~# _
```

Рисунок 32 – Запуск сервиса

Убедимся, что сервис запустился. Это показано на рисунке 33.

```
root@katya:~# systemctl show -p WantedBy NetworkManager
WantedBy=multi-user.target
root@katya:~#
```

Рисунок 33 – Запущенный сервис

ЧАСТЬ 2. ВИРТУАЛЬНЫЕ СЕТЕВЫЕ ИНТЕРФЕЙСЫ. СКРИПТЫ УПРАВЛЕНИЯ СЕРВИСАМИ.

Цель работы

Овладение навыками управления виртуальными сетевыми интерфейсами. Изучение команд управления сетевыми интерфейсами. Изучение синтаксиса и основных операторов командного интерпретатора

Ход работы

Для начала определим из параметров сетевого интерфейса eth0 последний разряд IP адреса N. Для этого воспользуемся командой, показанной на рисунке 34.

```
main@katya:~$ ip -4 addr show eth0
2: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN group
default qlen 1000
    inet 10.0.3.15/24 brd 10.0.3.255 scope global noprefixroute dynamic et
h0
        valid_lft 81703sec preferred_lft 81703sec
main@katya:~$
```

Рисунок 34 – Определение разряда адреса

В данном случае, этот разряд равен 15. Далее необходимо создать скрипт управления сервисом для активизации виртуального сетевого интерфейса, файл которого имеет название **3_4143_tegai.sh**. Листинг приведён на рисунках

Листинг программы 3_4143_tegai.sh

```
#!/bin/bash
INTERFACE="eth0:1"
IP_ADDRESS="10.0.3.15"
NETMASK='255.255.255.0'
case "$1" in
    start)
        > echo "Activating $INTERFACE with IP $IP_ADDRESS"
        > ifconfig eth0:1 $IP_ADDRESS netmask $NETMASK up
        > ;;
    stop)
        > echo "Deactivating"
        > ifconfig eth0:1 down
        > ;;
    restart)
        > echo "Restarting"
        > $0 stop
        > echo "Restarting route"
        > $0 start
        > ;;
    status)
        > echo "Network interfaces:"
        > ifconfig
        > echo "Static routes:"
        > route -n
        > ;;
    *)
```

Рисунок 35 – Листинг программы


```

> > ;;
> *)
> > echo "Usage: $0 {start|stop|restart|status}"
> > exit 1
> > ;;
esac
exit 0

```

Рисунок 36 – Листинг программы

Приступим к созданию файла скрипта управления службой. Листинг программы **3_4143_tegai.service** показан на рисунке 37.

```

[Unit]
Description=Network interface management for eth0:1
After=network.target

[Service]
Type=forking
ExecStart=/etc/init.d/3_4143_tegai.sh start
ExecStop=/etc/init.d/3_4143_tegai.sh stop
ExecReload=/etc/init.d/3_4143_tegai.sh restart
RemainAfterExit=true

[Install]
WantedBy=multi-user.target

```

Рисунок 37 – Листинг программы

Далее необходимо установить права на исполнение скрипта управления службой. Для этого нужно прописать команду, которая показана на рисунке

```

main@katya:~$ sudo chmod +x /etc/init.d/3_4143_tegai.sh

```

Рисунок 38 – Установка прав на исполнение скрипта

Теперь можно проверять написанный скрипт. Примеры показаны на рисунках 39 – 43.

```

main@katya:~$ sudo /etc/init.d/3_4143_tegai.sh status
Network interfaces:
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 84 bytes 5332 (5.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 84 bytes 5332 (5.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

Static routes:
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use If
ace

```

Рисунок 39 – Проверка работоспособности

```

main@katya:~$ sudo /etc/init.d/3_4143_tegai.sh start
Activating eth0:1 with IP 10.0.0.15

```

Рисунок 40 – Проверка работоспособности

```

main@katya:~$ sudo systemctl start 3_4143_tegai.service
Warning: 3_4143_tegai.service changed on disk. Run 'systemctl daemon-reload' to reload units.
main@katya:~$ sudo systemctl daemon-reload
main@katya:~$ sudo systemctl start 3_4143_tegai.service
main@katya:~$ sudo systemctl status 3_4143_tegai.service
• 3_4143_tegai.service - Network interface management for eth0:1
    Loaded: loaded (/etc/systemd/system/3_4143_tegai.service; disabled; ven
    Active: active (exited) since Thu 2025-02-27 19:56:04 MSK; 2min 27s ago
    Tasks: 0 (limit: 4915)
    CGroup: /system.slice/3_4143_tegai.service

feb 27 19:56:03 katya systemd[1]: Starting Network interface managment for
feb 27 19:56:04 katya 3_4143_tegai.sh[3005]: Activating eth0:1 with IP 10.
feb 27 19:56:04 katya systemd[1]: Started Network interface management for
main@katya:~$ sudo systemctl stop 3_4143_tegai.service

```

Рисунок 41 – Проверка работоспособности

```

main@katya:~$ sudo systemctl status 3_4143_tegai.service
• 3_4143_tegai.service - Network interface management for eth0:1
    Loaded: loaded (/etc/systemd/system/3_4143_tegai.service; disabled; ven
    Active: active (exited) since Thu 2025-02-27 20:16:31 MSK; 54min ago
    Process: 3289 ExecStart=/etc/init.d/3_4143_tegai.sh start (code=exited,
    Tasks: 0 (limit: 4915)
    CGroup: /system.slice/3_4143_tegai.service

feb 27 20:16:31 katya systemd[1]: Starting Network interface management fo
feb 27 20:16:31 katya 3_4143_tegai.sh[3289]: /etc/init.d/3_4143_tegai.sh:
feb 27 20:16:31 katya 3_4143_tegai.sh[3289]: Activating with IP 10.0.3.15
feb 27 20:16:31 katya 3_4143_tegai.sh[3289]: SIOCSIFFLAGS: Невозможно назн
feb 27 20:16:31 katya systemd[1]: Started Network interface management for
lines 1-12/12 (END)

```

Рисунок 42 – Проверка работоспособности

```

main@katya:~$ sudo systemctl restart 3_4143_tegai.service
main@katya:~$ sudo systemctl status 3_4143_tegai.service
• 3_4143_tegai.service - Network interface management for eth0:1
  Loaded: loaded (/etc/systemd/system/3_4143_tegai.service; disabled; ven
  Active: active (exited) since Thu 2025-02-27 21:11:50 MSK; 1s ago
  Process: 3455 ExecStop=/etc/init.d/3_4143_tegai.sh stop (code=exited, st
  Process: 3460 ExecStart=/etc/init.d/3_4143_tegai.sh start (code=exited,
feb 27 21:11:50 katya systemd[1]: Stopped Network interface management for
feb 27 21:11:50 katya systemd[1]: Starting Network interface management fo
feb 27 21:11:50 katya 3_4143_tegai.sh[3460]: /etc/init.d/3_4143_tegai.sh:
feb 27 21:11:50 katya 3_4143_tegai.sh[3460]: Activating with IP 10.0.3.15
feb 27 21:11:50 katya 3_4143_tegai.sh[3460]: SIOCSIFFLAGS: Невозможно назн
feb 27 21:11:50 katya systemd[1]: Started Network interface management for
lines 1-12/12 (END)

```

Рисунок 43 – Проверка работоспособности