

# ***Отчет по выполнению лабораторной работы***

***Лабораторная работа №5***

Полина Витальевна Барабаш

# ***Содержание***

<b><i>1</i></b>	<b><i>Цель работы</i></b>	<b><i>5</i></b>
<b><i>2</i></b>	<b><i>Выполнение лабораторной работы</i></b>	<b><i>6</i></b>
<b><i>3</i></b>	<b><i>Ответы на контрольные вопросы</i></b>	<b><i>20</i></b>
<b><i>4</i></b>	<b><i>Выводы</i></b>	<b><i>22</i></b>

## Список иллюстраций

2.1	Переход в режим суперпользователя . . . . .	6
2.2	Проверка статуса службы . . . . .	6
2.3	Установка службы . . . . .	7
2.4	Запуск службы и проверка статуса . . . . .	8
2.5	Добавление службы в автозапуск и проверка статуса . . . . .	8
2.6	Удаление службы из автозапуска и проверка статуса . . . . .	9
2.7	Вывод на экран символических ссылок, ответственных за запуск различных сервисов . . . . .	9
2.8	Повторное добавление службы в автозапуск и вывод символических ссылок . . . . .	10
2.9	Повторная проверка статуса службы . . . . .	10
2.10	Вывод списка зависимостей юнита . . . . .	11
2.11	Вывод списка юнитов, которые зависят от данного юнита . . . . .	11
2.12	Установка iptables . . . . .	12
2.13	Проверка статуса firewalld и iptables . . . . .	13
2.14	Запуск firewalld и iptables . . . . .	13
2.15	Просмотр firewalld.service . . . . .	14
2.16	Просмотр iptables.service . . . . .	15
2.17	Выгрузка службы и загрузка другой . . . . .	15
2.18	Запуск заблокированной службы . . . . .	16
2.19	Попытка добавить заблокированную службу в автозапуск . . . . .	16
2.20	Поиск списка целей, которые можно изолировать . . . . .	17
2.21	Вывод цели, установленной по умолчанию . . . . .	18
2.22	Выполнение команды <code>systemctl set-default multi-user.target</code> . . . . .	18
2.23	Выполнение команды <code>systemctl set-default graphical.target</code> . . . . .	19

## ***Список таблиц***

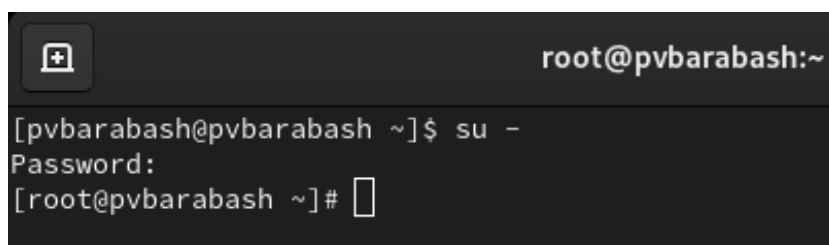
# ***1 Цель работы***

Получить навыки управления системными службами операционной системы посредством systemd.

## 2 Выполнение лабораторной работы

**Задание 1.** Получите полномочия администратора.

Я перешла в режим работы суперпользователя, используя команду `su -` (рис. [2.1]).

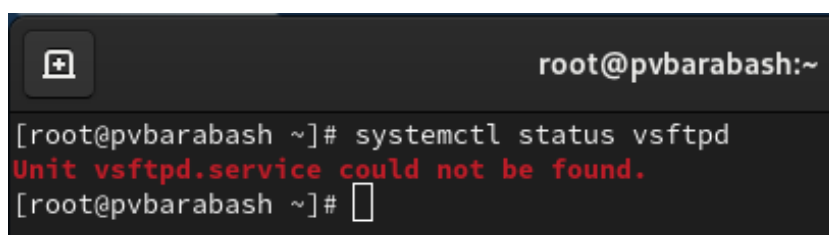
A terminal window with a dark background. The title bar shows a window icon and the text 'root@pvbarabash:~'. The terminal content shows the prompt '[pvbarabash@pvbarabash ~]\$' followed by the command 'su -'. Below this, the prompt changes to '[root@pvbarabash ~]#', indicating a successful transition to root privileges.

```
root@pvbarabash:~  
[pvbarabash@pvbarabash ~]$ su -  
Password:  
[root@pvbarabash ~]#
```

Рис. 2.1: Переход в режим суперпользователя

**Задание 2.** Проверьте статус службы Very Secure FTP.

Я ввела команду `systemctl status vsftpd`, чтобы проверить статус службы Very Secure FTP (рис. [2.2]).

A terminal window with a dark background. The title bar shows a window icon and the text 'root@pvbarabash:~'. The terminal content shows the prompt '[root@pvbarabash ~]#' followed by the command 'systemctl status vsftpd'. Below this, the output 'Unit vsftpd.service could not be found.' is displayed in red text. The prompt returns to '[root@pvbarabash ~]#'.

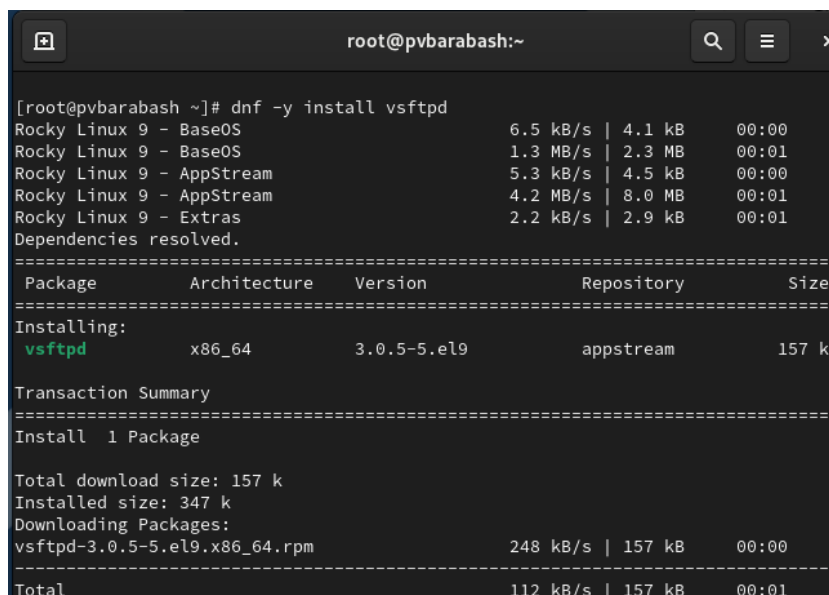
```
root@pvbarabash:~  
[root@pvbarabash ~]# systemctl status vsftpd  
Unit vsftpd.service could not be found.  
[root@pvbarabash ~]#
```

Рис. 2.2: Проверка статуса службы

Вывод команды показал, что сервис в настоящее время отключён, так как служба Very Secure FTP не установлена.

### Задание 3. Установите службу Very Secure FTP.

Я использовала команду `dnf -y install vsftpd`, чтобы установить службу Very Secure FTP (рис. [2.3]).



```
root@pvbarabash:~  
[root@pvbarabash ~]# dnf -y install vsftpd  
Rocky Linux 9 - BaseOS           6.5 kB/s | 4.1 kB    00:00  
Rocky Linux 9 - BaseOS           1.3 MB/s | 2.3 MB    00:01  
Rocky Linux 9 - AppStream        5.3 kB/s | 4.5 kB    00:00  
Rocky Linux 9 - AppStream        4.2 MB/s | 8.0 MB    00:01  
Rocky Linux 9 - Extras           2.2 kB/s | 2.9 kB    00:01  
Dependencies resolved.  
=====
```

Package	Architecture	Version	Repository	Size
Installing: vsftpd	x86_64	3.0.5-5.el9	appstream	157 k

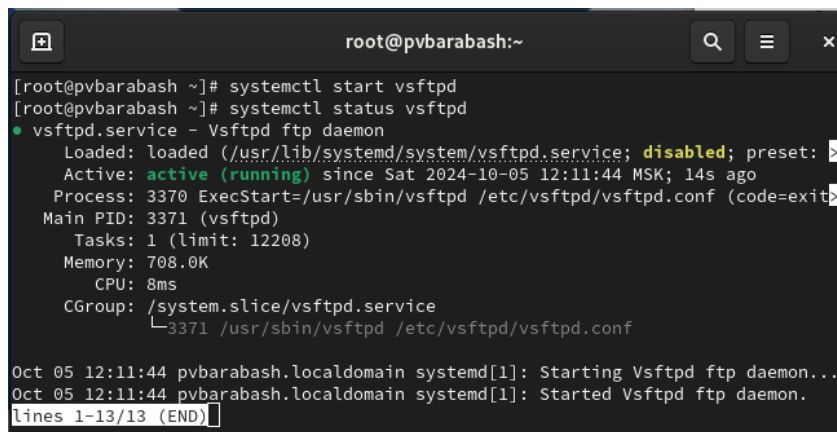
```
Transaction Summary  
=====
```

Install 1 Package			
Total download size: 157 k			
Installed size: 347 k			
Downloading Packages:			
vsftpd-3.0.5-5.el9.x86_64.rpm	248 kB/s	157 kB	00:00
-----			
Total	112 kB/s	157 kB	00:01

Рис. 2.3: Установка службы

### Задание 4. Запустите службу Very Secure FTP. Проверьте статус службы.

Я использовала команду `systemctl start vsftpd`, чтобы запустить службу Very Secure FTP. Затем я снова проверила статус службы, как можно видеть, служба в настоящее время работает, но не будет активирована при перезапуске операционной системы (рис. [2.4]).

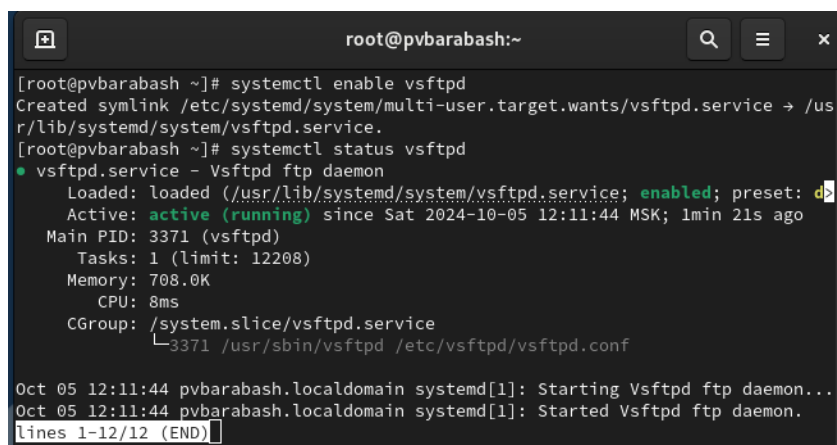


```
root@pvbarabash:~  
[root@pvbarabash ~]# systemctl start vsftpd  
[root@pvbarabash ~]# systemctl status vsftpd  
● vsftpd.service - Vsftpd ftp daemon  
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: >  
   Active: active (running) since Sat 2024-10-05 12:11:44 MSK; 14s ago  
     Process: 3370 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exit>  
    Main PID: 3371 (vsftpd)  
       Tasks: 1 (limit: 12208)  
      Memory: 708.0K  
         CPU: 8ms  
       CGroup: /system.slice/vsftpd.service  
               └─3371 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf  
  
Oct 05 12:11:44 pvbarabash.localdomain systemd[1]: Starting Vsftpd ftp daemon...  
Oct 05 12:11:44 pvbarabash.localdomain systemd[1]: Started Vsftpd ftp daemon.  
lines 1-13/13 (END)
```

Рис. 2.4: Запуск службы и проверка статуса

**Задание 5.** Добавьте службу Very Secure FTP в автозапуск при загрузке операционной системы, используя команду `systemctl enable`. Затем проверьте статус службы. Удалите службу из автозапуска, используя команду `systemctl disable`, и снова проверьте её статус.

Я использовала команду `systemctl enable vsftpd`, чтобы добавить службу в автозапуск, а затем проверила статус службы, как можно видеть, в информации приводится `enabled` (рис. [2.5]).



```
root@pvbarabash:~  
[root@pvbarabash ~]# systemctl enable vsftpd  
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /usr/lib/systemd/system/vsftpd.service.  
[root@pvbarabash ~]# systemctl status vsftpd  
● vsftpd.service - Vsftpd ftp daemon  
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: d>  
   Active: active (running) since Sat 2024-10-05 12:11:44 MSK; 1min 21s ago  
     Main PID: 3371 (vsftpd)  
       Tasks: 1 (limit: 12208)  
      Memory: 708.0K  
         CPU: 8ms  
       CGroup: /system.slice/vsftpd.service  
               └─3371 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf  
  
Oct 05 12:11:44 pvbarabash.localdomain systemd[1]: Starting Vsftpd ftp daemon...  
Oct 05 12:11:44 pvbarabash.localdomain systemd[1]: Started Vsftpd ftp daemon.  
lines 1-12/12 (END)
```

Рис. 2.5: Добавление службы в автозапуск и проверка статуса

Затем я удалила службу из автозапуска, используя команду `systemctl disable vsftpd` и вновь проверила статус, снова в информации `disabled` (рис. [2.6]).



```
root@pvbarabash:~  
[root@pvbarabash ~]# systemctl disable vsftpd  
Removed "/etc/systemd/system/multi-user.target.wants/vsftpd.service".  
[root@pvbarabash ~]# systemctl status vsftpd  
● vsftpd.service - Vsftpd ftp daemon  
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: ▶)  
   Active: active (running) since Sat 2024-10-05 12:11:44 MSK; 2min 2s ago  
   Main PID: 3371 (vsftpd)  
     Tasks: 1 (limit: 12208)  
    Memory: 708.0K  
       CPU: 8ms  
    CGroup: /system.slice/vsftpd.service  
            └─3371 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf  
  
Oct 05 12:11:44 pvbarabash.localdomain systemd[1]: Starting Vsftpd ftp daemon...  
Oct 05 12:11:44 pvbarabash.localdomain systemd[1]: Started Vsftpd ftp daemon.  
lines 1-12/12 (END)
```

Рис. 2.6: Удаление службы из автозапуска и проверка статуса

**Задание 6.** Выведите на экран символические ссылки, ответственные за запуск различных сервисов.

Я использовала команду `ls /etc/systemd/system/multi-user.target.wants`, чтобы вывести на экран символические ссылки, ответственные за запуск различных сервисов (рис. [2.7]).

```
root@pvbarabash:~  
[root@pvbarabash ~]# ls /etc/systemd/system/multi-user.target.wants  
atd.service          irqbalance.service  rsyslog.service  
auditd.service       kdump.service       smartd.service  
avahi-daemon.service libstoragemgmt.service sshd.service  
chronyd.service      mcelog.service      sssd.service  
crond.service         mdmonitor.service   tuned.service  
cups.path             ModemManager.service vboxadd.service  
cups.service          NetworkManager.service vboxadd-service.service  
firewalld.service     remote-fs.target     vmttoolsd.service  
[root@pvbarabash ~]#
```

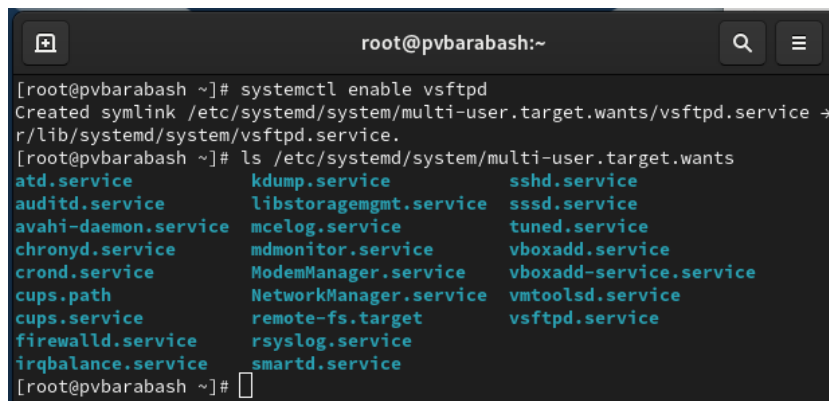
Рис. 2.7: Вывод на экран символических ссылок, ответственных за запуск различных сервисов

Как можно видеть, ссылки на `vsftpd.service` не существует.

**Задание 7.** Снова добавьте службу Very Secure FTP в автозапуск. Выведите на экран символические ссылки, ответственные за запуск различных сервисов.

Я снова добавила службу Very Secure FTP в автозапуск, используя команду `systemctl enable vsftpd`. А затем ввела команду из предыдущего задания, чтобы

вывести на экран символические ссылки, ответственные за запуск различных сервисов (рис. [2.8]).



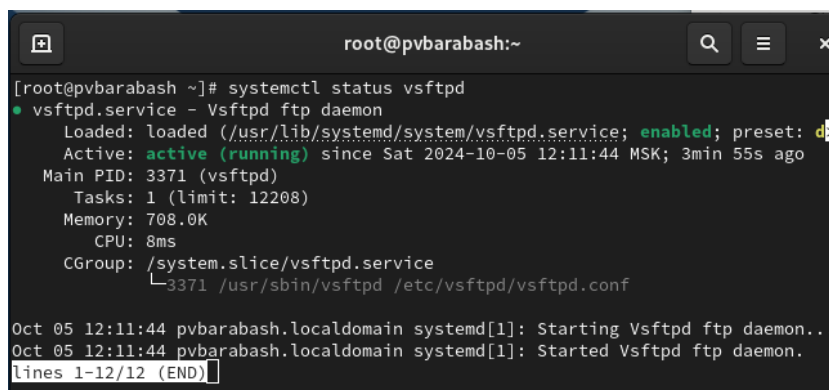
```
root@pvbarabash:~  
[root@pvbarabash ~]# systemctl enable vsftpd  
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service →  
r/lib/systemd/system/vsftpd.service.  
[root@pvbarabash ~]# ls /etc/systemd/system/multi-user.target.wants  
atd.service          kdump.service       sshd.service  
auditd.service       libstoragemgmt.service sssd.service  
avahi-daemon.service mcelog.service      tuned.service  
chronyd.service      mdmonitor.service   vboxadd.service  
crond.service        ModemManager.service vboxadd-service.service  
cups.path            NetworkManager.service vmtoolsd.service  
cups.service         remote-fs.target     vsftpd.service  
firewalld.service    rsyslog.service     smartd.service  
irqbalance.service  [root@pvbarabash ~]#
```

Рис. 2.8: Повторное добавление службы в автозапуск и вывод символических ссылок

Как можно видеть, файл vsftpd.service появился.

**Задание 8.** Снова проверьте статус службы Very Secure FTP.

Я снова проверила статус службы Very Secure FTP (рис. [2.9]).



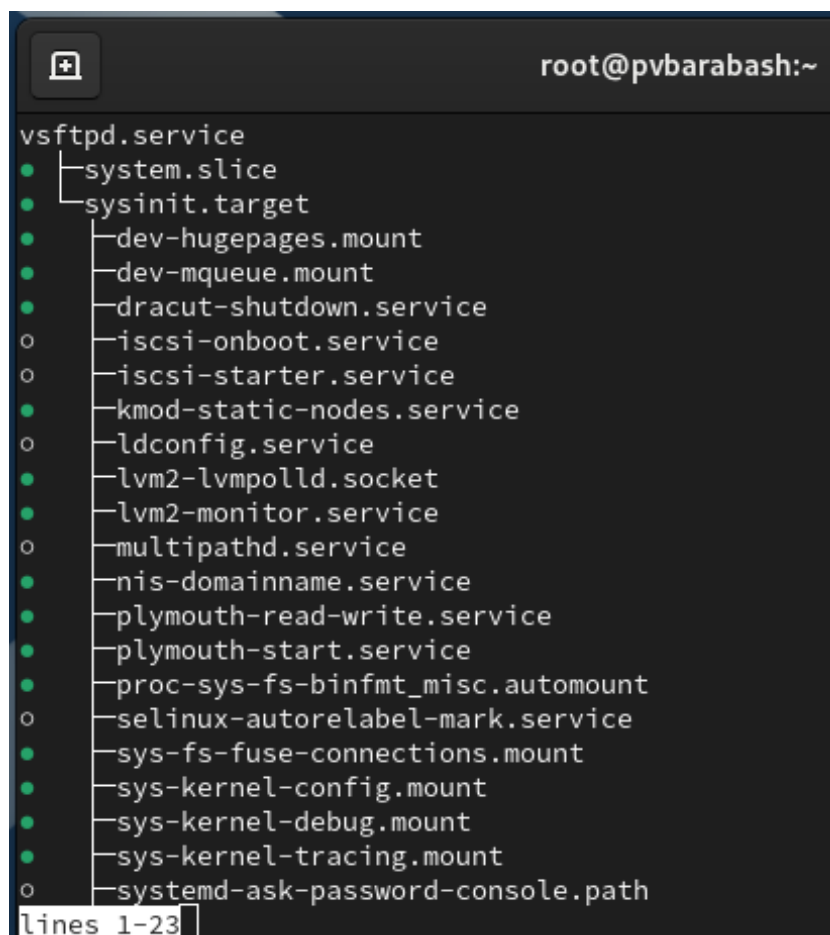
```
root@pvbarabash:~  
[root@pvbarabash ~]# systemctl status vsftpd  
● vsftpd.service - Vsftpd ftp daemon  
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: d  
   Active: active (running) since Sat 2024-10-05 12:11:44 MSK; 3min 55s ago  
   Main PID: 3371 (vsftpd)  
     Tasks: 1 (limit: 12208)  
    Memory: 708.0K  
       CPU: 8ms  
   CGroup: /system.slice/vsftpd.service  
           └─3371 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf  
  
Oct 05 12:11:44 pvbarabash.localdomain systemd[1]: Starting Vsftpd ftp daemon...  
Oct 05 12:11:44 pvbarabash.localdomain systemd[1]: Started Vsftpd ftp daemon.  
lines 1-12/12 (END)
```

Рис. 2.9: Повторная проверка статуса службы

Можно видеть, что для файла юнита состояние изменено с disabled на enabled.

**Задание 9.** Выведите на экран список зависимостей юнита.

Я использовала команду `systemctl list-dependencies vsftpd`, чтобы вывести на экран список зависимостей юнита (рис. [2.10]).

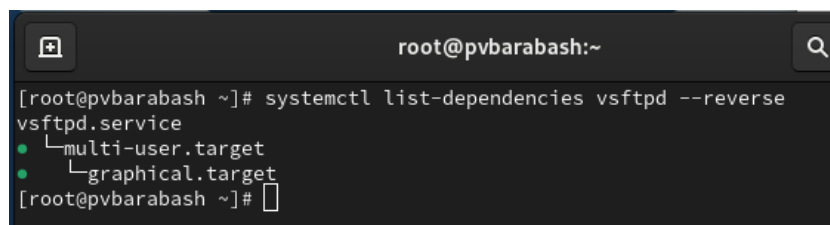


```
root@pvbarabash:~  
vsftpd.service  
├─system.slice  
├─sysinit.target  
├─dev-hugepages.mount  
├─dev-mqueue.mount  
├─dracut-shutdown.service  
├─iscsi-onboot.service  
├─iscsi-starter.service  
├─kmod-static-nodes.service  
├─ldconfig.service  
├─lvm2-lvmpolld.socket  
├─lvm2-monitor.service  
├─multipathd.service  
├─nis-domainname.service  
├─plymouth-read-write.service  
├─plymouth-start.service  
├─proc-sys-fs-binfmt_misc.automount  
├─selinux-autorelabel-mark.service  
├─sys-fs-fuse-connections.mount  
├─sys-kernel-config.mount  
├─sys-kernel-debug.mount  
├─sys-kernel-tracing.mount  
└─systemd-ask-password-console.path  
lines 1-23
```

Рис. 2.10: Вывод списка зависимостей юнита

**Задание 10.** Выведите на экран список юнитов, которые зависят от данного юнита.

Затем я использовала команду `systemctl list-dependencies vsftpd --reverse`, чтобы вывести на экран список юнитов, которые зависят от данного юнита (рис. [2.11]).

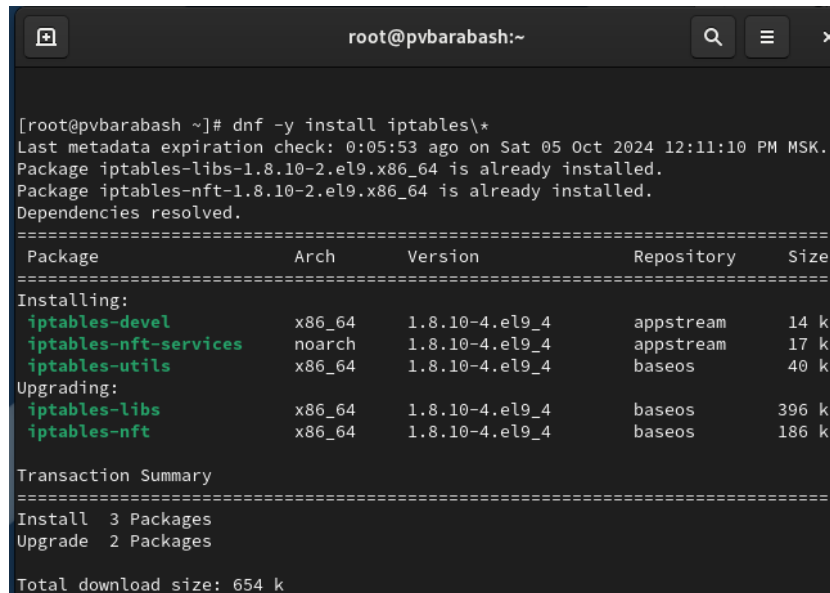


```
root@pvbarabash:~  
[root@pvbarabash ~]# systemctl list-dependencies vsftpd --reverse  
vsftpd.service  
├─multi-user.target  
└─graphical.target  
[root@pvbarabash ~]#
```

Рис. 2.11: Вывод списка юнитов, которые зависят от данного юнита

### Задание 11. Установите iptables.

Я использовала команду `dnf -y install iptables*`, чтобы установить iptables (рис. [2.12]).



```
root@pvbarabash:~  
[root@pvbarabash ~]# dnf -y install iptables*  
Last metadata expiration check: 0:05:53 ago on Sat 05 Oct 2024 12:11:10 PM MSK.  
Package iptables-libs-1.8.10-2.el9.x86_64 is already installed.  
Package iptables-nft-1.8.10-2.el9.x86_64 is already installed.  
Dependencies resolved.  
=====
```

Package	Arch	Version	Repository	Size
Installing:				
iptables-devel	x86_64	1.8.10-4.el9_4	appstream	14 k
iptables-nft-services	noarch	1.8.10-4.el9_4	appstream	17 k
iptables-utils	x86_64	1.8.10-4.el9_4	baseos	40 k
Upgrading:				
iptables-libs	x86_64	1.8.10-4.el9_4	baseos	396 k
iptables-nft	x86_64	1.8.10-4.el9_4	baseos	186 k

```
Transaction Summary  
=====
```

Transaction Summary
Install 3 Packages
Upgrade 2 Packages

```
Total download size: 654 k
```

Рис. 2.12: Установка iptables

### Задание 12. Проверьте статус firewalld и iptables.

Я использовала команды `systemctl status firewalld` и `systemctl status iptables`, чтобы проверить статус firewalld и iptables (рис. [2.13]).

```
root@pvbarabash:~  
[root@pvbarabash ~]# systemctl status firewalld  
● firewalld.service - firewalld - dynamic firewall daemon  
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset>  
   Active: active (running) since Sat 2024-10-05 12:00:17 MSK; 17min ago  
     Docs: man:firewalld(1)  
    Main PID: 754 (firewalld)  
       Tasks: 2 (limit: 12208)  
    Memory: 5.6M  
       CPU: 916ms  
    CGroup: /system.slice/firewalld.service  
           └─754 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid  
  
Oct 05 12:00:15 pvbarabash.localdomain systemd[1]: Starting firewalld - dynamic>  
Oct 05 12:00:17 pvbarabash.localdomain systemd[1]: Started firewalld - dynamic >  
[root@pvbarabash ~]# systemctl status iptables  
○ iptables.service - IPv4 firewall with iptables  
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; preset>  
   Active: inactive (dead)  
lines 1-3/3 (END)
```

Рис. 2.13: Проверка статуса firewalld и iptables

**Задание 13.** Попробуйте запустить firewalld и iptables.

Я выполнила команды `systemctl start firewalld` и `systemctl start iptables`, а затем опять проверила статус `firewalld` и `iptables`, как можно видеть, теперь `firewalld` неактивна, а `iptables`, которую мы запускали позже активна (рис. [2.14]).

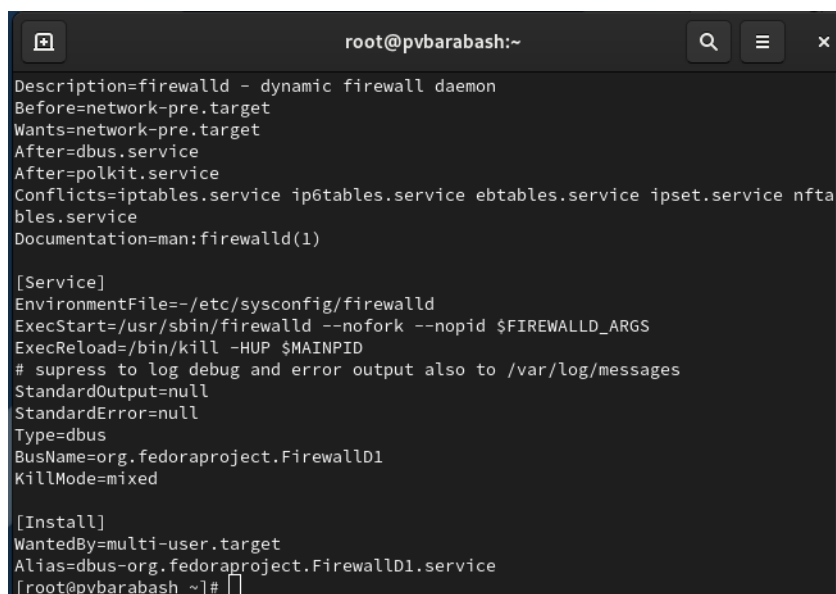
```
root@pvbarabash:~  
[root@pvbarabash ~]# systemctl start firewalld  
[root@pvbarabash ~]# systemctl start iptables  
[root@pvbarabash ~]# systemctl status firewalld  
○ firewalld.service - firewalld - dynamic firewall daemon  
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset>  
   Active: inactive (dead) since Sat 2024-10-05 12:19:04 MSK; 5s ago  
     Duration: 2.913s  
     Docs: man:firewalld(1)  
    Process: 3838 ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARG>  
   Main PID: 3838 (code=exited, status=0/SUCCESS)  
       CPU: 881ms  
  
Oct 05 12:19:00 pvbarabash.localdomain systemd[1]: Starting firewalld - dynamic>  
Oct 05 12:19:01 pvbarabash.localdomain systemd[1]: Started firewalld - dynamic >  
Oct 05 12:19:04 pvbarabash.localdomain systemd[1]: Stopping firewalld - dynamic>  
Oct 05 12:19:04 pvbarabash.localdomain systemd[1]: firewalld.service: Deactivat>  
Oct 05 12:19:04 pvbarabash.localdomain systemd[1]: Stopped firewalld - dynamic >  
[root@pvbarabash ~]# systemctl status iptables  
● iptables.service - IPv4 firewall with iptables  
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; preset>  
   Active: active (exited) since Sat 2024-10-05 12:19:04 MSK; 21s ago  
     Process: 3909 ExecStart=/usr/libexec/iptables/iptables.init start (code=exi>  
   Main PID: 3909 (code=exited, status=0/SUCCESS)
```

Рис. 2.14: Запуск firewalld и iptables

**Задание 14.** Введите `cat /usr/lib/systemd/system/firewalld.service` и опишите

настройки конфликтов для этого юнита при наличии.

Я ввела нужную команду и увидела следующие настройки конфликтов: iptables.service, ip6tables.service, ebtables.service, ipset.service, nftables.service (рис. [2.15]).



```
root@pvbarabash:~
Description=firewalld - dynamic firewall daemon
Before=network-pre.target
Wants=network-pre.target
After=dbus.service
After=polkit.service
Conflicts=iptables.service ip6tables.service ebtables.service ipset.service nfta
bles.service
Documentation=man:firewalld(1)

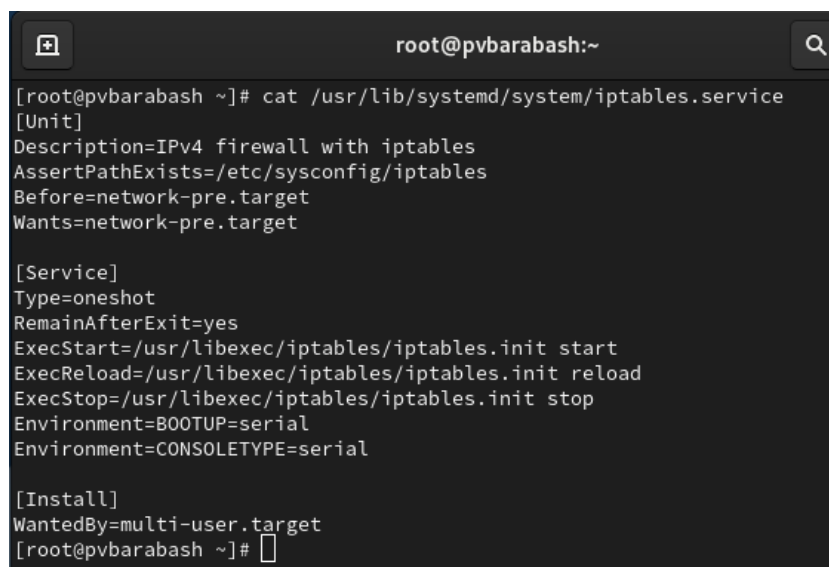
[Service]
EnvironmentFile=/etc/sysconfig/firewalld
ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS
ExecReload=/bin/kill -HUP $MAINPID
# suppress to log debug and error output also to /var/log/messages
StandardOutput=null
StandardError=null
Type=dbus
BusName=org.fedoraproject.FirewallD1
KillMode=mixed

[Install]
WantedBy=multi-user.target
Alias=dbus-org.fedoraproject.FirewallD1.service
[root@pvbarabash ~]#
```

Рис. 2.15: Просмотр firewalld.service

**Задание 15.** Введите `cat /usr/lib/systemd/system/iptables.service` и опишите настройки конфликтов для этого юнита.

Я ввела нужную команду, в данном случае строки о конфликтах нет (рис. [2.16]).

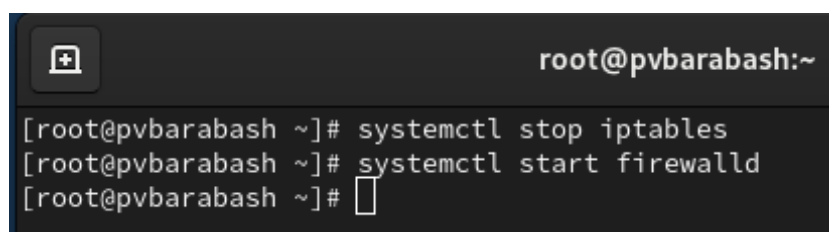


```
root@pvbarabash:~  
[root@pvbarabash ~]# cat /usr/lib/systemd/system/iptables.service  
[Unit]  
Description=IPv4 firewall with iptables  
AssertPathExists=/etc/sysconfig/iptables  
Before=network-pre.target  
Wants=network-pre.target  
  
[Service]  
Type=oneshot  
RemainAfterExit=yes  
ExecStart=/usr/libexec/iptables/iptables.init start  
ExecReload=/usr/libexec/iptables/iptables.init reload  
ExecStop=/usr/libexec/iptables/iptables.init stop  
Environment=BOOTUP=serial  
Environment=CONSOLETYPE=serial  
  
[Install]  
WantedBy=multi-user.target  
[root@pvbarabash ~]#
```

Рис. 2.16: Просмотр iptables.service

**Задание 16.** Выгрузите службу iptables (на всякий случай, чтобы убедиться, что данная служба не загружена в систему) и загрузите службу firewalld.

Я использовала команду `systemctl stop iptables`, чтобы выгрузить службу iptables, а затем использовала команду `systemctl start firewalld`, чтобы загрузить службу firewalld (рис. [2.17]).

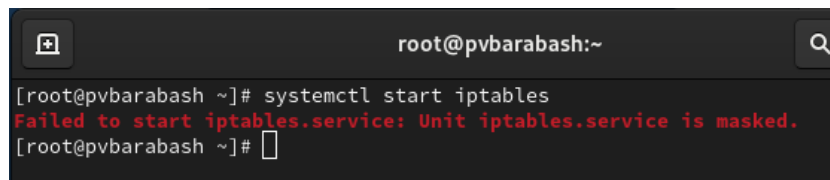


```
root@pvbarabash:~  
[root@pvbarabash ~]# systemctl stop iptables  
[root@pvbarabash ~]# systemctl start firewalld  
[root@pvbarabash ~]#
```

Рис. 2.17: Выгрузка службы и загрузка другой

**Задание 17.** Заблокируйте запуск iptables. Попробуйте запустить iptables.

Я использовала команду `systemctl mask iptables`, чтобы заблокировать запуск iptables. Затем я попробовала запустить iptables, используя команду `systemctl start iptables` и получила сообщение о невозможности, так как запуск заблокирован (рис. [2.18]).

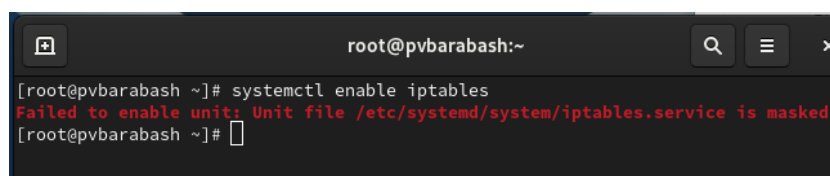


```
root@pvbarabash:~  
[root@pvbarabash ~]# systemctl start iptables  
Failed to start iptables.service: Unit iptables.service is masked.  
[root@pvbarabash ~]#
```

Рис. 2.18: Запуск заблокированной службы

**Задание 18.** Попробуйте добавить iptables в автозапуск.

Я использовала команду `systemctl enable iptables`, чтобы попробовать добавить iptables в автозапуск (рис. [2.19]).



```
root@pvbarabash:~  
[root@pvbarabash ~]# systemctl enable iptables  
Failed to enable unit: Unit file /etc/systemd/system/iptables.service is masked.  
[root@pvbarabash ~]#
```

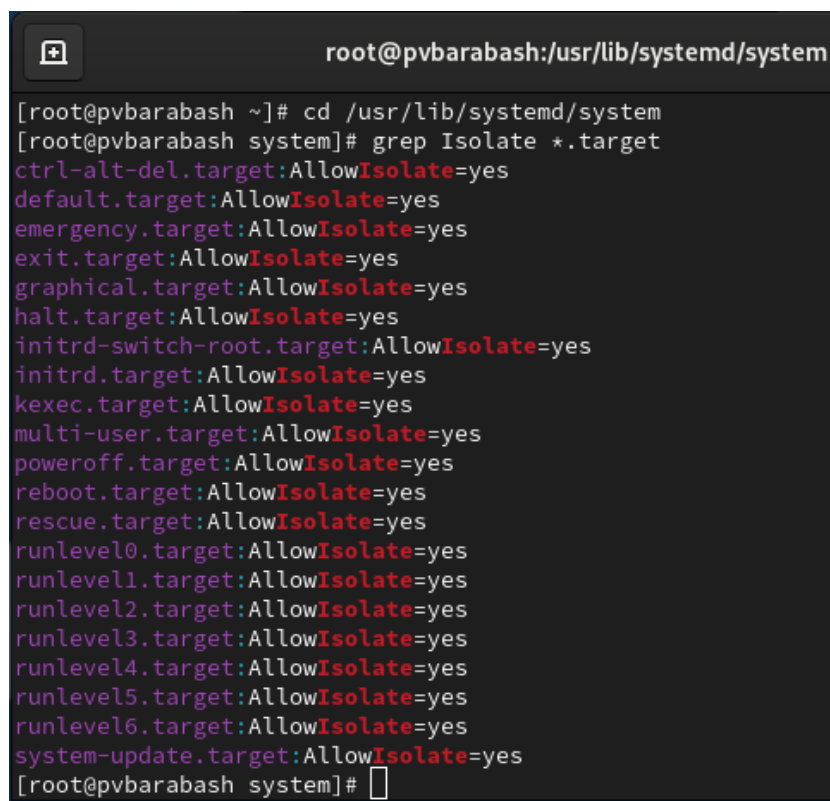
Рис. 2.19: Попытка добавить заблокированную службу в автозапуск

Как можно видеть, мы снова получили сообщение о невозможности этого действия.

**Задание 19.** Перейдите в каталог `systemd` и найдите список всех целей, которые можно изолировать.

Я перешла в каталог `/usr/lib/systemd/system` с помощью команды `cd`, а затем выполнила команду `grep Isolate *.target`, чтобы получить список всех целей, которые можно изолировать (рис. [2.20]).





```
root@pvbarabash:/usr/lib/systemd/system

[root@pvbarabash ~]# cd /usr/lib/systemd/system
[root@pvbarabash system]# grep Isolate *.target
ctrl-alt-del.target:AllowIsolate=yes
default.target:AllowIsolate=yes
emergency.target:AllowIsolate=yes
exit.target:AllowIsolate=yes
graphical.target:AllowIsolate=yes
halt.target:AllowIsolate=yes
initrd-switch-root.target:AllowIsolate=yes
initrd.target:AllowIsolate=yes
kexec.target:AllowIsolate=yes
multi-user.target:AllowIsolate=yes
poweroff.target:AllowIsolate=yes
reboot.target:AllowIsolate=yes
rescue.target:AllowIsolate=yes
runlevel0.target:AllowIsolate=yes
runlevel1.target:AllowIsolate=yes
runlevel2.target:AllowIsolate=yes
runlevel3.target:AllowIsolate=yes
runlevel4.target:AllowIsolate=yes
runlevel5.target:AllowIsolate=yes
runlevel6.target:AllowIsolate=yes
system-update.target:AllowIsolate=yes
[root@pvbarabash system]#
```

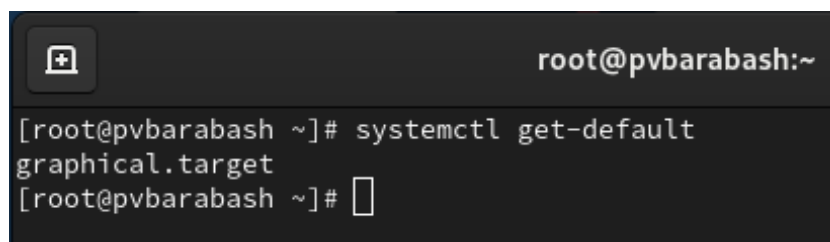
Рис. 2.20: Поиск списка целей, которые можно изолировать

**Задание 20.** Переключите операционную систему в режим восстановления. Перезапустите операционную систему следующим образом `systemctl isolate reboot.target`.

Я выполнила команду `systemctl isolate rescue.target`, чтобы перейти в режим восстановления, появился черный экран и мне было предложено ввести пароль пользователя, я ввела его, а затем ввела команду `systemctl isolate reboot.target` и операционная система перезагрузилась.

**Задание 21.** Выведите на экран цель, установленную по умолчанию.

Я ввела команду `systemctl get-default`, чтобы вывести на экран цель, установленную по умолчанию (рис. [2.21])

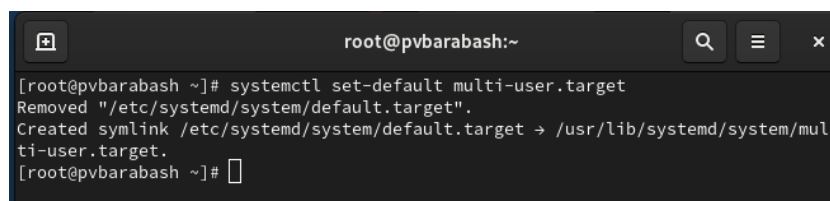


```
root@pvbarabash:~  
[root@pvbarabash ~]# systemctl get-default  
graphical.target  
[root@pvbarabash ~]#
```

Рис. 2.21: Вывод цели, установленной по умолчанию

**Задание 22.** Для запуска по умолчанию текстового режима введите `systemctl set-default multi-user.target`. Перезагрузите систему командой `reboot`. Убедитесь, что система загрузилась в текстовом режиме. Получите полномочия администратора. Для запуска по умолчанию графического режима введите `systemctl set-default graphical.target`. Вновь перезагрузите систему командой `reboot`. Убедитесь, что система загрузилась в графическом режиме.

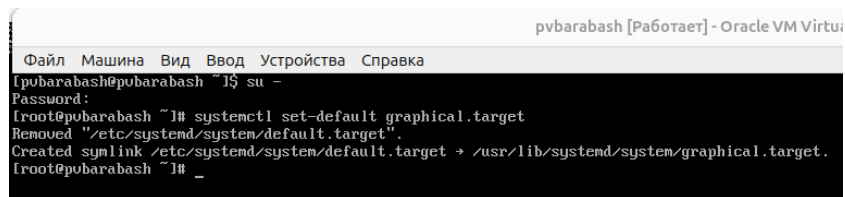
Я выполнила команду `systemctl set-default multi-user.target`, чтобы по умолчанию запускался текстовый режим (рис. [2.22]).



```
root@pvbarabash:~  
[root@pvbarabash ~]# systemctl set-default multi-user.target  
Removed "/etc/systemd/system/default.target".  
Created symlink /etc/systemd/system/default.target → /usr/lib/systemd/system/multi-user.target.  
[root@pvbarabash ~]#
```

Рис. 2.22: Выполнение команды `systemctl set-default multi-user.target`

Затем я перезагрузила систему командой `reboot`. Система загрузилась в текстовом режиме, я получила полномочия администратора и для запуска по умолчанию графического режима ввела `systemctl set-default graphical.target` (рис. [2.23]).



```
pvbarabash [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
pvbarabash@pvbarabash ~]$ su -
Password:
[root@pvbarabash ~]# systemctl set-default graphical.target
Removed "/etc/systemd/system/default.target".
Created symlink /etc/systemd/system/default.target → /usr/lib/systemd/system/graphical.target.
[root@pvbarabash ~]# _
```

Рис. 2.23: Выполнение команды `systemctl set-default graphical.target`

Затем я снова перезагрузила систему с помощью `reboot` и система загрузилась в графическом режиме.

### **3 Ответы на контрольные вопросы**

1. Что такое юнит (unit)? Приведите примеры. Юнит (unit) — это основной объект управления в системе инициализации systemd. Юниты могут представлять собой различные ресурсы, такие как службы, цели, таймеры, сокеты и т.д. Примеры юнитов:
  - Служба (service): `httpd.service` — служба веб-сервера Apache.
  - Цель (target): `multi-user.target` — цель для многопользовательского режима.
  - Таймер (timer): `my-timer.timer` — пользовательский таймер для запуска задачи.
2. Какая команда позволяет вам убедиться, что цель больше не входит в список автоматического запуска при загрузке системы? Нужно использовать команду `systemctl disable`.
3. Какую команду вы должны использовать для отображения всех сервисных юнитов, которые в настоящее время загружены? Нужно использовать команду `systemctl list-units -type=service`.
4. Как создать потребность (wants) в сервисе? Чтобы создать потребность (wants) в сервисе, необходимо создать или изменить файл юнита и добавить следующую строку в секцию [Install]: `Wants=` Затем выполнить команду для активации юнита: `systemctl enable`
5. Как переключить текущее состояние на цель восстановления (rescue target)? Использовать команду `systemctl isolate rescue.target`
6. Поясните причину получения сообщения о том, что цель не может быть

изолирована. Сообщение о том, что цель не может быть изолирована, может возникнуть, если цель имеет зависимости или если она не может быть активирована из-за текущего состояния системы (например, если другие юниты требуют ее активного состояния).

7. Вы хотите отключить службу `systemd`, но, прежде чем сделать это, вы хотите узнать, какие другие юниты зависят от этой службы. Какую команду вы бы использовали? Я бы использовала команду `systemctl list-dependencies systemd`.

## ***4 Выводы***

Я получила навыки управления системными службами операционной системы посредством systemd.