

Отчёт по лабораторной работе №5

Управление системными службами

Анна Саенко

Содержание

1	Цель работы	5
2	Ход выполнения работы	6
2.1	Управление службой vsftpd	6
2.2	Конфликты юнитов: iptables и firewalld	8
2.3	Изолируемые цели	12
2.4	Цель по умолчанию	13
3	Контрольные вопросы	15
4	Заключение	17

Список иллюстраций

2.1	Проверка отсутствия службы vsftpd и её установка	6
2.2	Запуск и проверка работы службы vsftpd	7
2.3	Работа с каталогом multi-user.target.wants	8
2.4	Просмотр зависимостей службы vsftpd	8
2.5	Установка пакетов iptables	9
2.6	Проверка состояния служб firewalld и iptables	9
2.7	Запуск служб firewalld и iptables	10
2.8	Попытка запуска и добавления в автозапуск замаскированного iptables	12
2.9	Просмотр изолируемых целей	12
2.10	Переход в режим восстановления и перезагрузка	13
2.11	Проверка цели по умолчанию	13
2.12	Изменение целей по умолчанию	14

Список таблиц

1 Цель работы

Получить навыки управления системными службами операционной системы посредством systemd.

2 Ход выполнения работы

2.1 Управление службой vsftpd

Сначала я получила права администратора, выполнив команду `su -`. После этого проверила статус службы **vsftpd** с помощью `systemctl status vsftpd`, однако система выдала сообщение, что такой сервис отсутствует.

```
aasaenko@aasaenko:~$ su
Password:
root@aasaenko:/home/aasaenko# systemctl status vsftpd
Unit vsftpd.service could not be found.
root@aasaenko:/home/aasaenko# systemctl status vsftpd
Unit vsftpd.service could not be found.
root@aasaenko:/home/aasaenko# dnf -y install vsftpd
Rocky Linux 10 - BaseOS                               12 kB/s | 3.9 kB  00:00
Rocky Linux 10 - BaseOS                               28 MB/s | 18 MB  00:00
Rocky Linux 10 - AppStream                             12 kB/s | 3.9 kB  00:00
Rocky Linux 10 - AppStream                            137 kB/s | 2.1 MB  00:15
Rocky Linux 10 - Extras                               9.3 kB/s | 3.1 kB  00:00
Rocky Linux 10 - Extras                               16 kB/s | 4.9 kB  00:00
Dependencies resolved.
=====
Package      Architecture Version      Repository      Size
=====
Installing:
vsftpd      x86_64      3.0.5-9.el10 appstream       170 k
Transaction Summary
=====
Install 1 Package

Total download size: 170 k
Installed size: 348 k
Downloading Packages:
vsftpd-3.0.5-9.el10.x86_64.rpm                        2.9 MB/s | 170 kB  00:00
-----
Total                                          565 kB/s | 170 kB  00:00
Running transaction check
Transaction check succeeded.
```

Рис. 2.1: Проверка отсутствия службы vsftpd и её установка

Для решения я установила пакет с FTP-сервером командой `dnf -y install vsftpd`. После завершения установки снова обратилась к службе и запустила её с помощью `systemctl start vsftpd`. Проверка через `systemctl status vsftpd` показала, что служба работает, но при этом находится в состоянии *disabled*.

```

root@aasaenko:/home/aasaenko#
root@aasaenko:/home/aasaenko# systemctl start vsftpd
root@aasaenko:/home/aasaenko# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
   Active: active (running) since Sun 2025-09-21 12:28:31 MSK; 2s ago
  Invocation: 965e197976c44a7398868b56a3dda6bb
    Process: 3904 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exited, status=0/SUCCESS)
   Main PID: 3905 (vsftpd)
      Tasks: 1 (limit: 24779)
     Memory: 752K (peak: 1.1M)
        CPU: 2ms
      CGroup: /system.slice/vsftpd.service
             └─3905 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 21 12:28:31 aasaenko.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Sep 21 12:28:31 aasaenko.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@aasaenko:/home/aasaenko#

```

Рис. 2.2: Запуск и проверка работы службы vsftpd

Затем я добавила службу в автозапуск командой `systemctl enable vsftpd` и снова проверила её состояние. Теперь статус изменился на *enabled*. После этого я отключила службу из автозапуска с помощью `systemctl disable vsftpd`, что также отразилось при проверке её статуса.

```

root@aasaenko:/home/aasaenko# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
   Active: active (running) since Sun 2025-09-21 12:28:31 MSK; 1min 33s ago
  Invocation: 965e197976c44a7398868b56a3dda6bb
    Main PID: 3905 (vsftpd)
      Tasks: 1 (limit: 24779)
     Memory: 752K (peak: 1.1M)
        CPU: 2ms
      CGroup: /system.slice/vsftpd.service
             └─3905 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 21 12:28:31 aasaenko.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Sep 21 12:28:31 aasaenko.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@aasaenko:/home/aasaenko#

root@aasaenko:/home/aasaenko# systemctl disable vsftpd
Removed '/etc/systemd/system/multi-user.target.wants/vsftpd.service'.
root@aasaenko:/home/aasaenko# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
   Active: active (running) since Sun 2025-09-21 12:28:31 MSK; 1min 55s ago
  Invocation: 965e197976c44a7398868b56a3dda6bb
    Main PID: 3905 (vsftpd)
      Tasks: 1 (limit: 24779)
     Memory: 752K (peak: 1.1M)
        CPU: 2ms
      CGroup: /system.slice/vsftpd.service
             └─3905 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 21 12:28:31 aasaenko.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Sep 21 12:28:31 aasaenko.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@aasaenko:/home/aasaenko#

```

Далее я вывела список символических ссылок, расположенных в каталоге `/etc/systemd/system/multi-user.target.wants`. Изначально ссылки на `vsftpd.service` там не было. После повторного добавления в автозапуск (`systemctl enable vsftpd`) ссылка появилась, что подтверждает корректное

включение службы в список автозагрузки.

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

Рис. 2.3: Работа с каталогом multi-user.target.wants

В завершение я посмотрела зависимости службы vsftpd командой `systemctl list-dependencies vsftpd`, а также список юнитов, зависящих от неё (`systemctl list-dependencies vsftpd --reverse`).

```
○ |multipathd.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
○ |systemd-binfmt.service
○ |systemd-boot-random-seed.service
○ |systemd-conext.service
○ |systemd-firstboot.service
○ |systemd-hibernate-clear.service
○ |systemd-hwdb-update.service
○ |systemd-journal-catalog-update.service
● |systemd-journal-flush.service
root@aasaenko:/home/aasaenko# systemctl list-dependencies vsftpd --reverse
vsftpd.service
● |multi-user.target
● |graphical.target
root@aasaenko:/home/aasaenko#
```

Рис. 2.4: Просмотр зависимостей службы vsftpd

2.2 Конфликты юнитов: iptables и firewalld

Сначала я получила права администратора и установила пакет **iptables** с помощью команды:


```
dnf -y install iptables\*.
```

```
root@aasaenko:/home/aasaenko# dnf -y install iptables\*
Last metadata expiration check: 0:06:27 ago on Sun 21 Sep 2025 12:27:09 PM MSK.
Package iptables-libs-1.8.11-8.el10_0.x86_64 is already installed.
Package iptables-nft-1.8.11-8.el10_0.x86_64 is already installed.
Dependencies resolved.
=====
Package                        Architecture      Version           Repository        Size
=====
Installing:
iptables-devel                x86_64            1.8.11-8.el10_0   appstream          17 k
iptables-nft-services         noarch            1.8.11-8.el10_0   appstream          24 k
iptables-utils                x86_64            1.8.11-8.el10_0   appstream          42 k
=====

Transaction Summary
=====
Install 3 Packages

Total download size: 82 k
Installed size: 142 k
Downloading Packages:
(1/3): iptables-devel-1.8.11-8.el10_0.x86_64.rpm           15 kB/s | 17 kB    00:01
(2/3): iptables-nft-services-1.8.11-8.el10_0.noarch.rpm    21 kB/s | 24 kB    00:01
(3/3): iptables-utils-1.8.11-8.el10_0.x86_64.rpm           37 kB/s | 42 kB    00:01
-----
Total                                                         52 kB/s | 82 kB    00:01
Running transaction check
Transaction check succeeded.
Running transaction test
```

Рис. 2.5: Установка пакетов iptables

Затем я проверила состояние служб `firewalld` и `iptables`. Первая оказалась активной и включённой, в то время как вторая была в состоянии *inactive (dead)*.

```
root@aasaenko:/home/aasaenko# systemctl status firewalld.service
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
   Active: active (running) since Sun 2025-09-21 12:24:22 MSK; 9min ago
 Invocation: 594eba0b524e49e58c0b8fc7ff53604b
    Docs: man:firewalld(1)
   Main PID: 953 (firewalld)
     Tasks: 2 (limit: 24779)
    Memory: 49.1M (peak: 51.1M)
       CPU: 176ms
    CGroup: /system.slice/firewalld.service
           └─953 /usr/bin/python3 -sP /usr/sbin/firewalld --nofork --nopid

Sep 21 12:24:22 aasaenko.localdomain systemd[1]: Starting firewalld.service - firewalld - dynamic firewall daemon:
Sep 21 12:24:22 aasaenko.localdomain systemd[1]: Started firewalld.service - firewalld - dynamic firewall daemon.
root@aasaenko:/home/aasaenko# systemctl status iptables.service
○ iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; preset: disabled)
   Active: inactive (dead)

root@aasaenko:/home/aasaenko#
```

Рис. 2.6: Проверка состояния служб firewalld и iptables

Далее я попыталась запустить обе службы последовательно. При запуске одной службы вторая завершала работу, что подтверждает их конфликт. В итоге `iptables` удалось активировать только после остановки `firewalld`.

```
root@aasaenko:/home/aasaenko# systemctl start firewalld
root@aasaenko:/home/aasaenko# systemctl start iptables
root@aasaenko:/home/aasaenko# systemctl status firewalld.service
○ firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
   Active: inactive (dead) since Sun 2025-09-21 12:35:34 MSK; 4s ago
     Duration: 11min 11.673s
  Invocation: 594eba0b524e49e58c0b8fc7ff53604b
       Docs: man:firewalld(1)
    Process: 953 ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS (code=exited, status=0/SUCCESS)
   Main PID: 953 (code=exited, status=0/SUCCESS)
    Mem peak: 51.1M
       CPU: 190ms

Sep 21 12:24:22 aasaenko.localdomain systemd[1]: Starting firewalld.service - firewalld - dynamic firewall daemon:
Sep 21 12:24:22 aasaenko.localdomain systemd[1]: Started firewalld.service - firewalld - dynamic firewall daemon:
Sep 21 12:35:34 aasaenko.localdomain systemd[1]: Stopping firewalld.service - firewalld - dynamic firewall daemon:
Sep 21 12:35:34 aasaenko.localdomain systemd[1]: firewalld.service: Deactivated successfully.
Sep 21 12:35:34 aasaenko.localdomain systemd[1]: Stopped firewalld.service - firewalld - dynamic firewall daemon:
root@aasaenko:/home/aasaenko# systemctl status iptables.service
● iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; preset: disabled)
   Active: active (exited) since Sun 2025-09-21 12:35:34 MSK; 8s ago
     Invocation: b7d47641b60f4b228f7b8dfff9a7a5d0
    Process: 5560 ExecStart=/usr/libexec/iptables/iptables.init start (code=exited, status=0/SUCCESS)
   Main PID: 5560 (code=exited, status=0/SUCCESS)
    Mem peak: 1.9M
       CPU: 7ms
```

Рис. 2.7: Запуск служб firewalld и iptables

Чтобы разобраться в причинах, я просмотрела содержимое юнитов `/usr/lib/systemd/system/firewalld.service` и `/usr/lib/systemd/system/iptables.service`. В конфигурации `firewalld` явно указано, что он конфликтует с сервисами `iptables.service`, `ip6tables.service`, `ebtables.service`, `ipset.service`. Для юнита `iptables.service` подобных ограничений прописано не было.

```

root@aasaenko:/home/aasaenko# cat /usr/lib/systemd/system/firewalld.service
[Unit]
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
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
DevicePolicy=closed
KeyringMode=private
LockPersonality=yes
MemoryDenyWriteExecute=yes
PrivateDevices=yes
ProtectClock=yes
ProtectControlGroups=yes
ProtectHome=yes
ProtectHostname=yes
ProtectKernelLogs=yes

root@aasaenko:/home/aasaenko# 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@aasaenko:/home/aasaenko#

```

После этого я вручную остановила службу iptables и запустила firewalld. Чтобы исключить возможность случайного запуска iptables, я замаскировала его командой `systemctl mask iptables`. В результате в каталоге `/etc/systemd/system/` была создана символическая ссылка на `/dev/null`.

Затем я убедилась, что запускать замаскированный сервис невозможно: при попытке выполнить `systemctl start iptables` система сообщила об ошибке. Аналогичная ситуация возникла и при добавлении сервиса в автозапуск — вывод

показал, что он замаскирован.

```
root@aasaenko:/home/aasaenko#  
root@aasaenko:/home/aasaenko# systemctl stop iptables.service  
root@aasaenko:/home/aasaenko# systemctl start firewalld.service  
root@aasaenko:/home/aasaenko# systemctl mask iptables.service  
Created symlink '/etc/systemd/system/iptables.service' → '/dev/null'.  
root@aasaenko:/home/aasaenko# systemctl start iptables  
Failed to start iptables.service: Unit iptables.service is masked.  
root@aasaenko:/home/aasaenko# systemctl enable iptables  
Failed to enable unit: Unit /etc/systemd/system/iptables.service is masked  
root@aasaenko:/home/aasaenko#
```

Рис. 2.8: Попытка запуска и добавления в автозапуск замаскированного iptables

2.3 Изолируемые цели

Сначала я получила права администратора и перешла в каталог `/usr/lib/systemd/system`, где выполнила команду `grep Isolate *.target`.

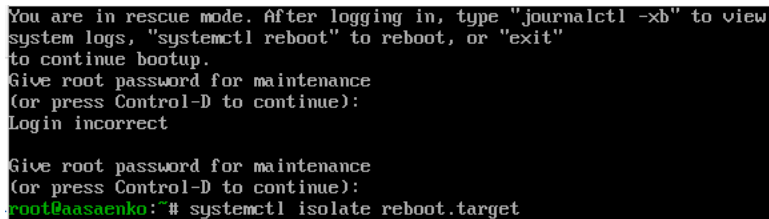
Вывод показал список всех целей, которые могут быть изолированы — для них установлено значение `AllowIsolate=yes`.

```
root@aasaenko:/home/aasaenko# cd /usr/lib/systemd/system  
root@aasaenko:/usr/lib/systemd/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  
soft-reboot.target:AllowIsolate=yes  
system-update.target:AllowIsolate=yes  
root@aasaenko:/usr/lib/systemd/system#
```

Рис. 2.9: Просмотр изолируемых целей

Затем я переключила систему в режим восстановления с помощью команды `systemctl isolate rescue.target`.

Для входа потребовалось ввести пароль суперпользователя. После этого я выполнила `systemctl isolate reboot.target`, чтобы инициировать перезагрузку.



```
You are in rescue mode. After logging in, type "journalctl -xb" to view
system logs, "systemctl reboot" to reboot, or "exit"
to continue bootup.
Give root password for maintenance
(or press Control-D to continue):
Login incorrect

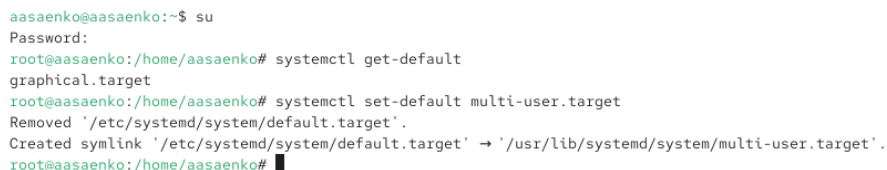
Give root password for maintenance
(or press Control-D to continue):
root@aasaenko:~# systemctl isolate reboot.target
```

Рис. 2.10: Переход в режим восстановления и перезагрузка

2.4 Цель по умолчанию

Я проверила, какая цель установлена в системе по умолчанию, с помощью команды `systemctl get-default`.

Вывод показал, что загружается `graphical.target`.



```
aasaenko@aasaenko:~$ su
Password:
root@aasaenko:/home/aasaenko# systemctl get-default
graphical.target
root@aasaenko:/home/aasaenko# 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@aasaenko:/home/aasaenko#
```

Рис. 2.11: Проверка цели по умолчанию

Затем я изменила цель по умолчанию на текстовый режим (`multi-user.target`) командой:

`systemctl set-default multi-user.target`.

После перезагрузки система загрузилась в текстовом режиме.

Далее я снова установила загрузку в графическом режиме, выполнив:
`systemctl set-default graphical.target`.
После перезагрузки система стартовала в графической оболочке.

```
Rocky Linux 10.0 (Red Quartz)
Kernel 6.12.0-55.12.1.el10_0.x86_64 on x86_64

Web console: https://aasaenko.localdomain:9090/ or https://10.0.2.15:9090/

aasaenko login: root
Password:
Last login: Sun Sep 21 12:42:57 on pts/0
root@aasaenko:~# 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@aasaenko:~#
```

Рис. 2.12: Изменение целей по умолчанию

3 Контрольные вопросы

1. Что такое юнит (unit)? Приведите примеры.

Юнит — это базовый объект управления в systemd, описывающий службу, сокет, устройство, точку монтирования или цель.

Примеры юнитов:

- `sshd.service` — служба SSH;
- `network.target` — цель для сетевых сервисов;
- `home.mount` — юнит для монтирования файловой системы.

2. Какая команда позволяет вам убедиться, что цель больше не входит в список автоматического запуска при загрузке системы?

Для этого используется команда `systemctl disable <имя_цели>`.

Проверить можно с помощью `systemctl is-enabled <имя_цели>`.

3. Какую команду вы должны использовать для отображения всех сервисных юнитов, которые в настоящее время загружены?

`systemctl list-units --type=service`

4. Как создать потребность (wants) в сервисе?

Для этого применяется команда `systemctl enable <имя_сервиса>`.

Она создаёт символическую ссылку в каталоге `*.wants/`, указывающую на соответствующий юнит.

5. **Как переключить текущее состояние на цель восстановления (rescue target)?**

Используется команда `systemctl isolate rescue.target`.

6. **Поясните причину получения сообщения о том, что цель не может быть изолирована.**

Это происходит, если в конфигурации юнита отсутствует параметр `AllowIsolate=yes`. Такие цели не предназначены для прямого запуска и не могут быть активированы командой `isolate`.

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?**

Для этого используется команда `systemctl list-dependencies <имя_сервиса> --reverse`.

4 Заключение

В ходе выполнения лабораторной работы я изучила принципы управления службами и целями в системе systemd.

Были выполнены следующие действия:

- установка и запуск службы vsftpd, её добавление и удаление из автозагрузки;
- просмотр зависимостей службы и анализ символических ссылок;
- исследование конфликтов между сервисами firewalld и iptables, демонстрация невозможности их одновременной работы;
- маскирование сервиса iptables для предотвращения случайного запуска;
- работа с изолируемыми целями, переключение системы в режим восстановления и перезагрузки;
- изменение целей загрузки по умолчанию между текстовым и графическим режимами.

В процессе работы я закрепила знания о типах юнитов systemd, их конфигурации и взаимодействии между ними. Полученный опыт показал, как управлять службами, обеспечивать их совместимость и контролировать режимы загрузки системы, что является важным навыком для системного администратора.