

Отчёт по лабораторной работе №5

Управление системными службами

Лабси Мохаммед

Содержание

1 Цель работы	5
2 Ход выполнения	6
2.1 Управление сервисами	6
2.2 Конфликты юнитов	10
2.3 Изолируемые цели systemd	13
2.4 Цель по умолчанию	15
2.5 Контрольные вопросы	16
3 Заключение	19

Список иллюстраций

2.1 Установка сервиса vsftpd с помощью dnf	7
2.2 Запуск и проверка статуса сервиса vsftpd	7
2.3 Включение и отключение автозапуска сервиса vsftpd	8
2.4 Статус сервиса vsftpd после повторного включения автозапуска	9
2.5 Просмотр прямых и обратных зависимостей сервиса vsftpd	10
2.6 Проверка статуса firewalld и iptables	11
2.7 Просмотр конфигурации юнита firewalld	12
2.8 Просмотр конфигурации юнита iptables	12
2.9 Маскирование сервиса iptables и попытки его запуска	13
2.10 Список изолируемых целей systemd	14
2.11 Переход системы в режим восстановления rescue.target	15
2.12 Установка multi-user.target в качестве цели по умолчанию	16
2.13 Установка graphical.target в качестве цели по умолчанию	16

Список таблиц

1 Цель работы

Получить навыки управления системными службами операционной системы посредством systemd.

2 Ход выполнения

2.1 Управление сервисами

1. Получены полномочия администратора с помощью команды **su -**.
После ввода пароля открыт сеанс суперпользователя *root*, что позволило управлять системными службами.
2. Выполнена проверка состояния службы **Very Secure FTP** с помощью команды

systemctl status vsftpd.

В результате получено сообщение о том, что юнит **vsftpd.service** не найден, что свидетельствует об отсутствии установленного сервиса в системе.

3. Установлен сервис **Very Secure FTP** с использованием пакетного менеджера командой

dnf -y install vsftpd.

В процессе установки были разрешены зависимости, загружен и установлен пакет *vsftpd* из репозитория *appstream*.

```

root@mlabsi:/etc/yum.repos.d# systemctl status vsftpd
Unit vsftpd.service could not be found.
root@mlabsi:/etc/yum.repos.d# dnf -y install vsftpd
Last metadata expiration check: 0:18:09 ago on Fri 12 Dec 2025 07:05:55 PM MSK.
Dependencies resolved.
=====
Package           Architecture      Version       Repository      Size
=====
Installing:
vsftpd            x86_64          3.0.5-10.el10   appstream     170 k
Transaction Summary
=====
Install 1 Package

Total download size: 170 k
Installed size: 344 k
Downloading Packages:
vsftpd-3.0.5-10.el10.x86_64.rpm          2.2 MB/s | 170 kB  00:00
Total                                         126 kB/s | 170 kB  00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing           :
Installing         : vsftpd-3.0.5-10.el10.x86_64
Running scriptlet: vsftpd-3.0.5-10.el10.x86_64
                               1/1
                               1/1
                               1/1

Installed:
vsftpd-3.0.5-10.el10.x86_64

Complete!
root@mlabsi:/etc/yum.repos.d#

```

Рис. 2.1: Установка сервиса vsftpd с помощью dnf

4. Служба **vsftpd** была запущена командой **systemctl start vsftpd**.

После запуска выполнена повторная проверка статуса.

5. Проверка состояния службы с помощью **systemctl status vsftpd** показала, что сервис находится в состоянии *active (running)*, однако параметр загрузки при старте системы имеет значение *disabled*, то есть служба не будет автоматически запущена после перезагрузки.

```

root@mlabsi:/etc/yum.repos.d# systemctl start vsftpd
root@mlabsi:/etc/yum.repos.d# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
   Active: active (running) since Fri 2025-12-12 19:24:35 MSK; 8s ago
     Invocation: 9954ebef190d4056a18f039782f1ec48
   Process: 8111 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exited, status=0/SUCCESS)
   Main PID: 8112 (vsftpd)
      Tasks: 1 (limit: 23140)
     Memory: 852K (peak: 1.2M)
        CPU: 2ms
      CGroup: /system.slice/vsftpd.service
              └─8112 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Dec 12 19:24:35 mlabsi.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Dec 12 19:24:35 mlabsi.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@mlabsi:/etc/yum.repos.d#

```

Рис. 2.2: Запуск и проверка статуса сервиса vsftpd

6. Для добавления службы **vsftpd** в автозапуск выполнена команда **systemctl**

enable vsftpd.

В результате была создана символическая ссылка в каталоге `/etc/systemd/system/multi-user.target.wants/`.

Повторная проверка статуса показала изменение состояния загрузки с `disabled` на `enabled`.

После этого служба была удалена из автозапуска с помощью команды `systemctl disable vsftpd`, что подтвердилось повторным выводом статуса.

```
root@mlabsi:/etc/yum.repos.d# systemctl enable vsftpd
Created symlink '/etc/systemd/system/multi-user.target.wants/vsftpd.service' → '/usr/lib/systemd/system/vsftpd.service'.
root@mlabsi:/etc/yum.repos.d# systemctl status vsftpd
● vsftpd service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
   Active: active (running) since Fri 2025-12-12 19:24:35 MSK; 45s ago
     Invocation: 9954ebef190d4056a18f039782f1ec48
   Main PID: 8112 (vsftpd)
      Tasks: 1 (limit: 23140)
     Memory: 852K (peak: 1.2M)
        CPU: 2ms
       CGroup: /system.slice/vsftpd.service
           └─8112 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Dec 12 19:24:35 mlabsi.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Dec 12 19:24:35 mlabsi.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@mlabsi:/etc/yum.repos.d#
root@mlabsi:/etc/yum.repos.d# systemctl disable vsftpd
Removed '/etc/systemd/system/multi-user.target.wants/vsftpd.service'.
root@mlabsi:/etc/yum.repos.d# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
   Active: active (running) since Fri 2025-12-12 19:24:35 MSK; 1min 9s ago
     Invocation: 9954ebef190d4056a18f039782f1ec48
   Main PID: 8112 (vsftpd)
      Tasks: 1 (limit: 23140)
     Memory: 852K (peak: 1.2M)
        CPU: 2ms
       CGroup: /system.slice/vsftpd.service
           └─8112 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Dec 12 19:24:35 mlabsi.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Dec 12 19:24:35 mlabsi.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@mlabsi:/etc/yum.repos.d#
```

Рис. 2.3: Включение и отключение автозапуска сервиса vsftpd

7. Выполнен просмотр списка символьических ссылок, отвечающих за запуск сервисов на уровне `multi-user.target`, командой
`ls /etc/systemd/system/multi-user.target.wants/`.

В выводе команды отсутствовала ссылка на `vsftpd.service`, что подтверждает корректное отключение автозапуска.

8. Служба `vsftpd` повторно добавлена в автозапуск командой `systemctl enable vsftpd`.

После этого при повторном выводе содержимого каталога `multi-user.target.wants` появилась символическая ссылка на файл `/usr/lib/systemd/system/vsftpd.service`.

9. Повторная проверка состояния службы с помощью **systemctl status vsftpd** показала, что сервис находится в состоянии *active (running)*, а параметр загрузки изменён на *enabled*.

```
root@mlabsi:/etc/yum.repos.d# ls /etc/systemd/system/multi-user.target.wants/
atd.service      cups.path        libstoragemgmt.service   remote-fs.target    vboxadd.service
auditd.service   cups.service     mdmonitor.service     rsyslog.service   vboxadd-service.service
audit-rules.service firewalld.service mdmonitor.service smartd.service   vmtoolsd.service
avahi-daemon.service httpd.service ModemManager.service sshd.service
chronynd.service  irqbalance.service NetworkManager.service sssd.service
crond.service    kdump.service   remote-cryptsetup.target tuned.service
root@mlabsi:/etc/yum.repos.d# systemctl enable vsftpd
Created symlink '/etc/systemd/system/multi-user.target.wants/vsftpd.service' → '/usr/lib/systemd/system/vsftpd.service'.
root@mlabsi:/etc/yum.repos.d# ls /etc/systemd/system/multi-user.target.wants/
atd.service      cups.path        libstoragemgmt.service   remote-fs.target    vboxadd.service
auditd.service   cups.service     mdmonitor.service     rsyslog.service   vboxadd-service.service
audit-rules.service firewalld.service mdmonitor.service smartd.service   vmtoolsd.service
avahi-daemon.service httpd.service ModemManager.service sshd.service
chronynd.service  irqbalance.service NetworkManager.service sssd.service
crond.service    kdump.service   remote-cryptsetup.target tuned.service
root@mlabsi:/etc/yum.repos.d# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded ('/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled')
   Active: active (running) since Fri 2025-12-12 19:24:35 MSK; 2min 9s ago
     Invocation: 9954ebef190d4056a18f039782f1ec48
   Main PID: 8112 (vsftpd)
      Tasks: 1 (limit: 23140)
     Memory: 852K (peak: 1.2M)
        CPU: 2ms
       CGroup: /system.slice/vsftpd.service
               └─8112 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Dec 12 19:24:35 mlabsi.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Dec 12 19:24:35 mlabsi.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@mlabsi:/etc/yum.repos.d#
```

Рис. 2.4: Статус сервиса vsftpd после повторного включения автозапуска

10. Выведен список зависимостей юнита **vsftpd** командой

systemctl list-dependencies vsftpd.

В результате отображены целевые юниты и сервисы, от которых зависит работа данного сервиса.

11. Выведен список юнитов, которые зависят от **vsftpd**, с использованием команды

systemctl list-dependencies vsftpd –reverse.

В выводе отображены цели *multi-user.target* и *graphical.target*, что указывает на использование сервиса на соответствующих уровнях загрузки.

```
└─ sys-kernel-tracing.mount
○ └─ systemd-ask-password-console.path
○ └─ systemd-binfmt.service
○ └─ systemd-boot-random-seed.service
○ └─ systemd-confext.service
○ └─ systemd-firstboot.service
○ └─ systemd-hibernate-clear.service
○ └─ systemd-hwdb-update.service
○ └─ systemd-journal-catalog-update.service
● └─ systemd-journal-flush.service
● └─ systemd-journald.service
○ └─ systemd-machine-id-commit.service
● └─ systemd-modules-load.service
● └─ systemd-network-generator.service
○ └─ systemd-pcrmachine.service
○ └─ systemd-pcrphase-sysinit.service
root@mlabsi:/etc/yum.repos.d#
root@mlabsi:/etc/yum.repos.d# systemctl list-dependencies vsftpd --reverse
vsftpd.service
●   └─multi-user.target
●     └─graphical.target
root@mlabsi:/etc/yum.repos.d# █
```

Рис. 2.5: Просмотр прямых и обратных зависимостей сервиса vsftpd

2.2 Конфликты юнитов

- Получены полномочия администратора и установлен пакет **iptables** с помощью команды

dnf -y install iptables*.

В процессе установки были обновлены и добавлены необходимые компоненты подсистемы фильтрации пакетов.

- Проверено состояние сервисов **firewalld** и **iptables** с помощью команд **systemctl status firewalld** и **systemctl status iptables**.

Служба *firewalld* находилась в состоянии *active (running)*, в то время как *iptables* была неактивна.

```

Upgraded:
  iptables-libs-1.8.11-11.el10.x86_64           iptables-nft-1.8.11-11.el10.x86_64
Installed:
  iptables-devel-1.8.11-11.el10.x86_64          iptables-nft-services-1.8.11-11.el10.noarch
  iptables-utils-1.8.11-11.el10.x86_64          kernel-modules-extra-matched-6.12.0-124.16.1.el10_1.x86_64

Complete!
root@mlabsi:/etc/yum.repos.d#
root@mlabsi:/etc/yum.repos.d# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-12-12 19:00:43 MSK; 28min ago
     Invocation: 021df2d22a5664abfb1eacf7e06f1583
      Docs: man:firewalld(1)
     Main PID: 1137 (firewalld)
       Tasks: 2 (limit: 23140)
      Memory: 50.5M (peak: 71.9M)
        CPU: 565ms
       CGroup: /system.slice/firewalld.service
                  └─1137 /usr/bin/python3 -sP /usr/sbin/firewalld --nofork --nrepid

Dec 12 19:00:40 mlabsi.localdomain systemd[1]: Starting firewalld.service - firewalld - dynamic firewall daemon...
Dec 12 19:00:43 mlabsi.localdomain systemd[1]: Started firewalld.service - firewalld - dynamic firewall daemon.
root@mlabsi:/etc/yum.repos.d# systemctl status iptables
○ iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; preset: disabled)
   Active: inactive (dead)
root@mlabsi:/etc/yum.repos.d# systemctl start firewalld
root@mlabsi:/etc/yum.repos.d# systemctl start iptables
root@mlabsi:/etc/yum.repos.d# 

```

Рис. 2.6: Проверка статуса firewalld и iptables

3. Выполнена попытка запуска обеих служб командами

systemctl start firewalld и **systemctl start iptables**.

При этом наблюдается конфликт: одновременная работа данных сервисов невозможна, так как они используют одни и те же механизмы фильтрации сетевых пакетов.

4. Выполнен просмотр файла юнита **firewalld** с помощью команды

cat /usr/lib/systemd/system/firewalld.service.

В описании юнита присутствует директива **Conflicts**, указывающая на несовместимость с сервисами *iptables*, *ip6tables*, *ebtables* и *ipset*.

```

root@mlabsi:/etc/yum.repos.d#
root@mlabsi:/etc/yum.repos.d# 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
ExecStartPost=/usr/bin/firewall-cmd --state
# don't fail ExecStartPost on RUNNING_BUT_FAILED
SuccessExitStatus=251
ExecReload=/bin/kill -HUP $MAINPID
StandardOutput=null
StandardError=null
Type=dbus
BusName=org.fedoraproject.FirewallD1
KillMode=mixed
DevicePolicy=closed
KeyringMode=private

```

Рис. 2.7: Просмотр конфигурации юнита firewalld

5. Выполнен просмотр файла юнита **iptables** командой

cat /usr/lib/systemd/system/iptables.service.

Данный юнит предназначен для инициализации правил iptables и привязывается к цели *multi-user.target*, что подтверждает его конфликт с *firewalld*.

```

root@mlabsi:/etc/yum.repos.d#
root@mlabsi:/etc/yum.repos.d# 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@mlabsi:/etc/yum.repos.d# █

```

Рис. 2.8: Просмотр конфигурации юнита iptables

6. Для исключения конфликта служба **iptables** была остановлена командой **systemctl stop iptables**, после чего служба **firewalld** была успешно запущена командой **systemctl start firewalld**.
 7. Для предотвращения случайного запуска сервиса **iptables** выполнена команда **systemctl mask iptables**.
В результате была создана символическая ссылка на */dev/null* для файла */etc/systemd/system/iptables.service*, что делает запуск сервиса невозможным.
8. Попытка запуска замаскированной службы с помощью **systemctl start iptables** завершилась ошибкой, указывающей на то, что сервис замаскирован и не может быть запущен.
 9. Попытка добавления сервиса **iptables** в автозапуск с помощью **systemctl enable iptables** также завершилась ошибкой, при этом статус службы отображается как *masked*, что подтверждает корректность выполненных действий.

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

Рис. 2.9: Маскирование сервиса iptables и попытки его запуска

2.3 Изолируемые цели systemd

1. Получены полномочия администратора и выполнен переход в каталог с системными юнитами **systemd** с помощью команды **cd /usr/lib/systemd/system**. Для определения целей, которые могут быть изолированы, выполнен поиск

директивы **AllowIsolate** в файлах целей командой **grep Isolate *.target**. В результате получен список целей, для которых параметр *AllowIsolate=yes*, что указывает на возможность их изоляции и установки в качестве цели по умолчанию. Среди них присутствуют *multi-user.target*, *graphical.target*, *rescue.target*, *reboot.target* и другие.

```
root@mlabsi:/etc/yum.repos.d# cd /usr/lib/systemd/system
root@mlabsi:/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@mlabsi:/usr/lib/systemd/system#
```

Рис. 2.10: Список изолируемых целей systemd

2. Выполнено переключение операционной системы в режим восстановления с помощью команды

systemctl isolate rescue.target.

После выполнения команды система перешла в *rescue mode*, при этом для продолжения работы потребовалось ввести пароль пользователя *root* на системной консоли.

```
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):
root@mlabsi:~# systemctl isolate reboot.target
```

Рис. 2.11: Переход системы в режим восстановления rescue.target

3. Выполнен перезапуск операционной системы путём изоляции цели **reboot.target** с помощью команды **systemctl isolate reboot.target**.

В результате система была корректно перезагружена.

2.4 Цель по умолчанию

4. После загрузки системы получены полномочия администратора и выполнена проверка цели, установленной по умолчанию, командой **systemctl get-default**.

В выводе команды указано, что текущей целью по умолчанию является *graphical.target*.

5. Для установки текстового режима загрузки по умолчанию выполнена команда

systemctl set-default multi-user.target.

В результате была удалена существующая символьическая ссылка */etc/systemd/system/default.target* и создана новая ссылка на файл */usr/lib/systemd/system/multi-user.target*.

```

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

```

Рис. 2.12: Установка multi-user.target в качестве цели по умолчанию

- После перезагрузки системы выполнен повторный вход с правами администратора и снова выполнена команда **systemctl get-default**.

Результат подтвердил, что система загрузилась в текстовом режиме с целью *multi-user.target*.

- Для возврата графического режима загрузки по умолчанию выполнена команда

systemctl set-default graphical.target.

В результате была создана символическая ссылка на файл */usr/lib/systemd/system/graphical.target*.

После очередной перезагрузки системы подтверждено, что загрузка происходит в графическом режиме.

```

Rocky Linux 8.0 (Red Quartz)
Kernel 6.12.0-55.39.1.el8_0.x86_64 on x86_64

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

mlabsi login: root
Password:
Last login: Fri Dec 12 19:37:12 on pts/0
root@mlabsi:~# systemctl get-default
graphical.target
root@mlabsi:~# 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@mlabsi:~# █

```

Рис. 2.13: Установка graphical.target в качестве цели по умолчанию

2.5 Контрольные вопросы

- Что такое юнит (unit)? Приведите примеры.**

Юнит — это основной объект управления в системе **systemd**, представляющий собой описание ресурса, службы или состояния системы.

Юниты определяют, как и когда должны запускаться различные компоненты операционной системы.

Примеры юнитов:

- **service** – сервисы и демоны (например, `vsftpd.service`, `sshd.service`);
- **target** – логические группы юнитов и уровни загрузки (например, `multi-user.target`, `graphical.target`);
- **mount** – точки монтирования файловых систем (например, `home.mount`);
- **socket** – сокеты для активации сервисов (например, `sshd.socket`).

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

Для проверки цели, установленной по умолчанию, используется команда `systemctl get-default`.

Если в выводе отсутствует проверяемая цель, значит она не используется для автоматического запуска при загрузке системы.

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

Для отображения всех загруженных сервисных юнитов используется команда

`systemctl list-units --type=service`.

Она выводит список активных и загруженных сервисов с указанием их текущего состояния.

4. Как создать потребность (wants) в сервисе?

Потребность (*Wants*) в сервисе создаётся путём добавления директивы **Wants=** в конфигурационный файл другого юнита

либо с помощью команды `systemctl enable`, которая создаёт соответствующую символьическую ссылку между юнитами.

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

Для перехода в режим восстановления используется команда **systemctl isolate rescue.target**.

При этом система останавливает большинство сервисов и запрашивает пароль пользователя *root* для продолжения работы.

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

Такое сообщение появляется, если в файле цели отсутствует директива **AllowIsolate=yes**.

Это означает, что данная цель не предназначена для изоляции и не может быть запущена отдельно со всеми своими зависимостями.

7. Вы хотите отключить службу systemd, но перед этим необходимо узнать, какие другие юниты зависят от неё. Какую команду следует использовать?

Для просмотра юнитов, зависящих от выбранного сервиса, используется команда

systemctl list-dependencies --reverse.

Она выводит список всех юнитов, для которых данный сервис является зависимостью.

3 Заключение

В ходе лабораторной работы были изучены принципы управления системными службами и целями в системе **systemd**. Освоены операции установки, запуска, остановки и добавления сервисов в автозапуск, а также анализ их состояний и зависимостей. На практике рассмотрены изолируемые цели, механизм переключения режимов работы системы и изменение цели загрузки по умолчанию. Кроме того, изучены конфликты юнитов на примере сервисов **firewalld** и **iptables** и методы предотвращения их одновременного запуска. Полученные навыки позволяют эффективно администрировать службы и управлять процессом загрузки операционной системы.