

Отчёт по лабораторной работе №5

Управление системными службами

Анастасия Мазуркевич

Содержание

| | | |
|----------|--------------------------------|-----------|
| 1 | Цель работы | 5 |
| 2 | Ход выполнения | 6 |
| 2.1 | Управление сервисами | 6 |
| 2.2 | Конфликты юнитов | 9 |
| 2.3 | Изолируемые цели | 12 |
| 2.4 | Цель по умолчанию | 13 |
| 3 | Контрольные вопросы | 15 |
| 4 | Заключение | 17 |

Список иллюстраций

| | | |
|------|--|----|
| 2.1 | Проверка статуса vsftpd и установка пакета | 6 |
| 2.2 | Запуск и проверка статуса vsftpd | 7 |
| 2.3 | Проверка наличия симлинка и добавление vsftpd в автозапуск . . | 8 |
| 2.4 | Зависимости vsftpd | 8 |
| 2.5 | Статус firewalld и iptables | 9 |
| 2.6 | Запуск firewalld и iptables | 10 |
| 2.7 | Конфигурация firewalld.service | 10 |
| 2.8 | Конфигурация iptables.service | 11 |
| 2.9 | Маскирование и блокировка iptables | 12 |
| 2.10 | Поиск изолируемых целей | 13 |
| 2.11 | Переключение в rescue и reboot target | 13 |
| 2.12 | Установка multi-user.target по умолчанию | 14 |
| 2.13 | Установка graphical.target по умолчанию | 14 |

Список таблиц

1 Цель работы

Получить навыки управления системными службами операционной системы посредством systemd.

2 Ход выполнения

2.1 Управление сервисами

Сначала были получены полномочия администратора с помощью команды `su`.

Проверка статуса службы **Very Secure FTP (vsftpd)** показала, что она отсутствует в системе:

```
admazurkevich@admazurkevich:~$ su
Password:
root@admazurkevich:/home/admazurkevich# systemctl status vsftpd
Unit vsftpd.service could not be found.
root@admazurkevich:/home/admazurkevich# dnf -y install vsftpd
Rocky Linux 10 - BaseOS                               5.9 kB/s | 3.9 kB  00:00
Rocky Linux 10 - BaseOS                               26 MB/s | 18 MB  00:00
Rocky Linux 10 - AppStream                             2.3 kB/s | 3.9 kB  00:01
Rocky Linux 10 - AppStream                             1.0 MB/s | 2.1 MB  00:01
Rocky Linux 10 - Extras                                11 kB/s | 3.1 kB  00:00
Rocky Linux 10 - Extras                                11 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                        973 kB/s | 170 kB  00:00
-----
Total
Running transaction check
Transaction check succeeded.
```

Рис. 2.1: Проверка статуса vsftpd и установка пакета

Для установки использовалась команда:

```
dnf -y install vsftpd
```

После успешной загрузки пакета служба была установлена в систему.

Далее был выполнен запуск службы:

```
systemctl start vsftpd
```

Повторная проверка показала, что сервис **работает**, однако его состояние оставалось **disabled** (он не запускался бы при старте ОС).

```
root@admazurkevich: /home/admazurkevich# systemctl start vsftpd
root@admazurkevich: /home/admazurkevich# 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 15:21:58 MSK; 6s ago
  Invocation: 9e7328f435864233943e6d54d9210004
    Process: 3561 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exited, status=0/SUCCESS)
   Main PID: 3562 (vsftpd)
      Tasks: 1 (limit: 24779)
     Memory: 748K (peak: 1.1M)
        CPU: 3ms
      CGroup: /system.slice/vsftpd.service
              └─3562 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 21 15:21:58 admazurkevich.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Sep 21 15:21:58 admazurkevich.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@admazurkevich: /home/admazurkevich#
```

Рис. 2.2: Запуск и проверка статуса vsftpd

Для добавления службы в автозагрузку использовалась команда:

```
systemctl enable vsftpd
```

Теперь в статусе службы отобразилось состояние **enabled**.

Затем через команду `systemctl disable vsftpd` демон был исключён из автозапуска, что также подтвердил статус.

```
root@admazurkevich: /home/admazurkevich# systemctl enable vsftpd
Created symlink '/etc/systemd/system/multi-user.target.wants/vsftpd.service' → '/usr/lib/systemd/system/vsftpd.service'.
root@admazurkevich: /home/admazurkevich# 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 15:21:58 MSK; 2min 29s ago
  Invocation: 9e7328f435864233943e6d54d9210004
   Main PID: 3562 (vsftpd)
      Tasks: 1 (limit: 24779)
     Memory: 748K (peak: 1.1M)
        CPU: 3ms
      CGroup: /system.slice/vsftpd.service
              └─3562 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 21 15:21:58 admazurkevich.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Sep 21 15:21:58 admazurkevich.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@admazurkevich: /home/admazurkevich#

root@admazurkevich: /home/admazurkevich# systemctl disable vsftpd
Removed '/etc/systemd/system/multi-user.target.wants/vsftpd.service'.
root@admazurkevich: /home/admazurkevich# 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 15:21:58 MSK; 3min 12s ago
  Invocation: 9e7328f435864233943e6d54d9210004
   Main PID: 3562 (vsftpd)
      Tasks: 1 (limit: 24779)
     Memory: 748K (peak: 1.1M)
        CPU: 3ms
      CGroup: /system.slice/vsftpd.service
              └─3562 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 21 15:21:58 admazurkevich.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Sep 21 15:21:58 admazurkevich.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@admazurkevich: /home/admazurkevich#
```

Символические ссылки, отвечающие за запуск сервисов в режиме multi-user, были просмотрены командой:

```
ls /etc/systemd/system/multi-user.target.wants/
```

Изначально ссылка на **vsftpd.service** отсутствовала. После повторного выполнения команды `systemctl enable vsftpd` она появилась в каталоге и указывала на юнит `/usr/lib/systemd/system/vsftpd.service`.

```

root@admazurkevich:/home/admazurkevich# ls /etc/systemd/system/multi-user.target.wants/
atd.service      cups.path        mcelog.service  rsyslog.service  vboxadd-service.service
auditd.service   cups.service     mdmmonitor.service  smartd.service   vmtoolsd.service
audit-rules.service  firewalld.service  ModemManager.service  sshd.service
avahi-daemon.service  irqbalance.service  NetworkManager.service  sssd.service
chronyd.service      kdump.service      remote-cryptsetup.target  tuned.service
crond.service        libstoragemgmt.service  remote-fs.target         vboxadd.service

root@admazurkevich:/home/admazurkevich# systemctl enable vsftpd
Created symlink '/etc/systemd/system/multi-user.target.wants/vsftpd.service' -> '/usr/lib/systemd/system/vsftpd.service'.

root@admazurkevich:/home/admazurkevich# ls /etc/systemd/system/multi-user.target.wants/
atd.service      cups.path        mcelog.service  rsyslog.service  vboxadd-service.service
auditd.service   cups.service     mdmmonitor.service  smartd.service   vmtoolsd.service
audit-rules.service  firewalld.service  ModemManager.service  sshd.service     vsftpd.service
avahi-daemon.service  irqbalance.service  NetworkManager.service  sssd.service
chronyd.service      kdump.service      remote-cryptsetup.target  tuned.service
crond.service        libstoragemgmt.service  remote-fs.target         vboxadd.service

```

Рис. 2.3: Проверка наличия симлинка и добавление vsftpd в автозапуск

Дополнительно были изучены зависимости сервиса.

Команда

```
systemctl list-dependencies vsftpd
```

показала, что демон связан с целевым юнитом **multi-user.target**.

Для просмотра обратных зависимостей использовалась команда:

```
systemctl list-dependencies vsftpd --reverse
```

Из вывода видно, что запуск vsftpd включён в состав multi-user.target и, косвенно, graphical.target.

```

● |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-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

root@admazurkevich:/home/admazurkevich# systemctl list-dependencies vsftpd --reverse
vsftpd.service
● |multi-user.target
● |graphical.target
root@admazurkevich:/home/admazurkevich#

```

Рис. 2.4: Зависимости vsftpd

2.2 Конфликты юнитов

Некоторые сервисы в Linux не могут работать одновременно, так как их функциональность пересекается. Примером являются службы **firewalld** и **iptables**: запуск одной автоматически деактивирует другую.

После получения полномочий администратора была установлена служба **iptables** с помощью команды:

```
dnf -y install iptables*
```

Проверка статуса показала:

- **firewalld** был активен и включён в автозагрузку.
- **iptables** находился в состоянии inactive.

```
root@admazurkevich:/home/admazurkevich# 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 15:20:07 MSK; 10min ago
     Invocation: 0c22e71ac1994a7692366deaba@d539c
       Docs: man:firewalld(1)
    Main PID: 951 (firewalld)
       Tasks: 2 (limit: 24779)
      Memory: 49.2M (peak: 51.2M)
         CPU: 252ms
    CGroup: /system.slice/firewalld.service
            └─951 /usr/bin/python3 -sP /usr/sbin/firewalld --nofork --nopid

Sep 21 15:20:07 admazurkevich.localdomain systemd[1]: Starting firewalld.service - firewalld - dynamic firewall daemon:
Sep 21 15:20:07 admazurkevich.localdomain systemd[1]: Started firewalld.service - firewalld - dynamic firewall daemon:
root@admazurkevich:/home/admazurkevich# 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@admazurkevich:/home/admazurkevich#
```

Рис. 2.5: Статус firewalld и iptables

Попытка запуска обеих служб показала конфликт: при активации **firewalld** демон **iptables** останавливался и наоборот.

```

root@admazurkevich:/home/admazurkevich# systemctl start firewalld
root@admazurkevich:/home/admazurkevich# systemctl start iptables
root@admazurkevich:/home/admazurkevich# systemctl status firewalld.service
O 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 15:31:22 MSK; 2s ago
  Duration: 11min 15.153s
  Invocation: 0c22e71ac1994a7692366deaba0d539c
  Docs: man:firewalld(1)
  Process: 951 ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS (code=exited, status=0/SUCCESS)
  Main PID: 951 (code=exited, status=0/SUCCESS)
  Mem peak: 51.2M
  CPU: 267ms

Sep 21 15:20:07 admazurkevich.localdomain systemd[1]: Starting firewalld.service - firewalld - dynamic firewall daemo
Sep 21 15:20:07 admazurkevich.localdomain systemd[1]: Started firewalld.service - firewalld - dynamic firewall daemo
Sep 21 15:31:22 admazurkevich.localdomain systemd[1]: Stopping firewalld.service - firewalld - dynamic firewall daemo
Sep 21 15:31:22 admazurkevich.localdomain systemd[1]: firewalld.service: Deactivated successfully.
Sep 21 15:31:22 admazurkevich.localdomain systemd[1]: Stopped firewalld.service - firewalld - dynamic firewall daemo
root@admazurkevich:/home/admazurkevich# 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 15:31:22 MSK; 5s ago
  Invocation: ff270ff42c034972b7bcca023374edd7
  Process: 5551 ExecStart=/usr/libexec/iptables/iptables.init start (code=exited, status=0/SUCCESS)
  Main PID: 5551 (code=exited, status=0/SUCCESS)
  Mem peak: 1.6M
  CPU: 8ms

Sep 21 15:31:22 admazurkevich.localdomain systemd[1]: Starting iptables.service - IPv4 firewall with iptables...
Sep 21 15:31:22 admazurkevich.localdomain iptables.init[5551]: iptables: Applying firewall rules: [ OK ]
Sep 21 15:31:22 admazurkevich.localdomain systemd[1]: Finished iptables.service - IPv4 firewall with iptables.
root@admazurkevich:/home/admazurkevich# q

```

Рис. 2.6: Запуск firewalld и iptables

Просмотр конфигурационных файлов юнитов подтвердил наличие конфликта. Файл `firewalld.service` содержит строку:

`Conflicts=iptables.service ip6tables.service ebtables.service ipset.service`

Это означает, что запуск **firewalld** несовместим с перечисленными сервисами.

```

root@admazurkevich:/home/admazurkevich# 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
# supress 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
ProtectKernelModules=no
ProtectKernelTunables=no
ProtectSystem=yes

```

Рис. 2.7: Конфигурация firewalld.service

Файл `iptables.service` не содержит явных конфликтов, однако запускается в том же целевом окружении `multi-user.target`, что и `firewalld`, что приводит к конкуренции при работе.

```
root@admazurkevich: /home/admazurkevich# 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@admazurkevich: /home/admazurkevich#
```

Рис. 2.8: Конфигурация `iptables.service`

Для корректной работы был остановлен `iptables` (`systemctl stop iptables`) и запущен `firewalld`. Чтобы исключить случайный запуск `iptables`, использовалась команда:

```
systemctl mask iptables
```

В результате для `/etc/systemd/system/iptables.service` была создана символическая ссылка на `/dev/null`, что полностью блокирует запуск данного сервиса.

Попытка запуска `iptables` после маскирования привела к ошибке, указывающей, что юнит замаскирован. Аналогично, при добавлении его в автозапуск (`systemctl enable iptables`) система сообщила, что запуск невозможен.

```

root@admazurkevich:/home/admazurkevich# systemctl stop iptables.service
root@admazurkevich:/home/admazurkevich# systemctl start firewalld.service
root@admazurkevich:/home/admazurkevich# systemctl mask iptables.service
Created symlink '/etc/systemd/system/iptables.service' -> '/dev/null'.
root@admazurkevich:/home/admazurkevich# systemctl start iptables
Failed to start iptables.service: Unit iptables.service is masked.
root@admazurkevich:/home/admazurkevich# systemctl enable iptables
Failed to enable unit: Unit /etc/systemd/system/iptables.service is masked
root@admazurkevich:/home/admazurkevich# systemctl status iptables
O iptables.service
   Loaded: masked (Reason: Unit iptables.service is masked.)
   Active: inactive (dead) since Sun 2025-09-21 15:35:13 MSK; 54s ago
 Duration: 3min 50.128s
 Invocation: ff270ff42c034972b7bcca023374edd7
   Main PID: 5551 (code=exited, status=0/SUCCESS)
    Mem peak: 2.1M
       CPU: 33ms

Sep 21 15:31:22 admazurkevich.localdomain systemd[1]: Starting iptables.service - IPv4 firewall with iptables...
Sep 21 15:31:22 admazurkevich.localdomain iptables.init[5551]: iptables: Applying firewall rules: [ OK ]
Sep 21 15:31:22 admazurkevich.localdomain systemd[1]: Finished iptables.service - IPv4 firewall with iptables.
Sep 21 15:35:12 admazurkevich.localdomain systemd[1]: Stopping iptables.service - IPv4 firewall with iptables...
Sep 21 15:35:13 admazurkevich.localdomain iptables.init[6160]: iptables: Setting chains to policy ACCEPT: raw mangle
Sep 21 15:35:13 admazurkevich.localdomain iptables.init[6160]: iptables: Flushing firewall rules: [ OK ]
Sep 21 15:35:13 admazurkevich.localdomain systemd[1]: iptables.service: Deactivated successfully.
Sep 21 15:35:13 admazurkevich.localdomain systemd[1]: Stopped iptables.service - IPv4 firewall with iptables.
root@admazurkevich:/home/admazurkevich#

```

Рис. 2.9: Маскирование и блокировка iptables

2.3 Изолируемые цели

В системе **systemd** некоторые цели (targets) могут быть изолированы. Это означает, что система переключается в данную цель вместе со всеми её зависимостями. Такие цели могут быть установлены и как цель по умолчанию.

После получения полномочий администратора был выполнен поиск всех целей с параметром AllowIsolate=yes:

```
grep Isolate *.target
```

Среди них: multi-user.target, graphical.target, rescue.target, reboot.target, poweroff.target и другие.

```

root@admazurkevich:/home/admazurkevich#
root@admazurkevich:/home/admazurkevich# cd /usr/lib/systemd/system
root@admazurkevich:/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@admazurkevich:/usr/lib/systemd/system#

```

Рис. 2.10: Поиск изолируемых целей

Далее система была переведена в режим восстановления с помощью:

```
systemctl isolate rescue.target
```

Для продолжения работы потребовался пароль суперпользователя.

Затем система была перезапущена командой:

```
systemctl isolate reboot.target
```

```

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@admazurkevich:~# systemctl isolate reboot.target _

```

Рис. 2.11: Переключение в rescue и reboot target

2.4 Цель по умолчанию

Определение текущей цели по умолчанию выполнялось командой:

```
systemctl get-default
```

Изначально это был graphical.target.

Затем цель по умолчанию была изменена на многопользовательский текстовый режим:

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

После перезапуска ОС система загрузилась без графической оболочки.

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

Рис. 2.12: Установка multi-user.target по умолчанию

Для возврата графической оболочки использовалась команда:

```
systemctl set-default graphical.target
```

После очередного перезапуска система загрузилась в графическом режиме.

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

Web console: https://admazurkevich.localdomain:9898/ or https://10.0.2.15:9898/

admazurkevich login: 123456
Password:
Login incorrect

admazurkevich login:
Password:
Login incorrect

admazurkevich login: root
Password:
Last login: Sun Sep 21 15:39:05 on pts/0
root@admazurkevich:~# systemctl get-default
multi-user.target
root@admazurkevich:~# 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@admazurkevich:~# _
```

Рис. 2.13: Установка graphical.target по умолчанию

3 Контрольные вопросы

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

Юнит (unit) — это объект в системе **systemd**, который описывает управляемый ресурс: службу, точку монтирования, устройство или цель.

Примеры:

- сервис: `sshd.service`, `vsftpd.service`;
- цель: `multi-user.target`, `graphical.target`;
- точка монтирования: `home.mount`;
- сокет: `cups.socket`.

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_юнита>
```

После отключения можно проверить отсутствие символической ссылки в каталоге `/etc/systemd/system/multi-user.target.wants/`.

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

Команда:

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

4. Как создать потребность (wants) в сервисе?

Необходимо добавить символическую ссылку на юнит в каталог `wants` соответствующей цели.

Пример:

```
systemctl enable <имя_сервиса>
```

Эта команда автоматически создаёт ссылку в каталоге `/etc/systemd/system/<цель>.wants/`.

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.

Если юнит-цель не содержит параметра `AllowIsolate=yes` в своём описании, то `systemd` запрещает её изоляцию.

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 просмотра зависимостей используется команда:

```
systemctl list-dependencies --reverse <имя_сервиса>
```

Она показывает все юниты, которые зависят от указанного.

4 Заключение

В ходе лабораторной работы были изучены механизмы управления сервисами и целями в системе **systemd**: установка и запуск служб, добавление и удаление их из автозагрузки, анализ зависимостей и конфликтов между юнитами. Также были рассмотрены изолируемые цели и изменение цели по умолчанию. Полученные навыки позволяют уверенно администрировать процессы и режимы работы операционной системы.