

# **Отчет о лабораторной работе**

**Лабораторная работа №5**

Казначеев Сергей Ильич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Контрольные вопросы</b>	<b>13</b>
<b>4</b>	<b>Выводы</b>	<b>15</b>

# Список иллюстраций

2.1	1	6
2.2	2	7
2.3	3	7
2.4	4	7
2.5	5	8
2.6	6	8
2.7	7	8
2.8	8	9
2.9	9	9
2.10	10	9
2.11	11	10
2.12	12	10
2.13	13	10
2.14	14	11
2.15	15	11
2.16	16	11
2.17	17	11
2.18	18	12
2.19	19	12
2.20	20	12

## **Список таблиц**

# 1 Цель работы

Получить навыки управления системными службами операционной системы посредством systemd

## 2 Выполнение лабораторной работы

Для начала откроем терминал и перейдем в супер пользователя root после чего проверяем статус службы Very Secure FTP и устанавливаем его (рис. 2.1).

```
[s1kaznacheev@localhost ~]$ su -
Пароль:
[root@localhost ~]# systemctl status vsftpd
Unit vsftpd.service could not be found.
[root@localhost ~]# dnf -y install vsftpd
Extra Packages for Enterprise Linux 9 - x86_64
^CExtra Packages for Enterprise Linux 9 - 88% [=====] 34 kB/s | 25 kB 00:00
Extra Packages for Enterprise Linux 9 - x86_64
Extra Packages for Enterprise Linux 9 - 88% [=====] 2.7 MB/s | 18 MB 00:00 ET
Extra Packages for Enterprise Linux 9 - x86_64
Extra Packages for Enterprise Linux 9 - 88% [=====] 496 kB/s | 18 MB 00:36
Ошибка: Не удалось загрузить метаданные для репозитория «epel»: Librepo was interrupted by a signal
[root@localhost ~]# dnf -y install vsftpd
Extra Packages for Enterprise Linux 9 - x86_64
Extra Packages for Enterprise Linux 9 - x86_64
Rocky Linux 9 - BaseOS
Rocky Linux 9 - BaseOS
Rocky Linux 9 - AppStream
Rocky Linux 9 - AppStream
Rocky Linux 9 - Extras
3.7 kB/s | 2.9 kB 00:00
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий  Размер
-----
Установка:
vsftpd     x86_64       3.0.5-6.el9 appstream    157 k
=====
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 157 k
Объем изменений: 347 k
Загрузка пакетов:
vsftpd-3.0.5-6.el9.x86_64.rpm
-----
313 kB/s | 157 kB 00:00
Общий размер
-----
181 kB/s | 157 kB 00:00
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка      :
Установка       : vsftpd-3.0.5-6.el9.x86_64 1/1
Запуск скрипта: vsftpd-3.0.5-6.el9.x86_64 1/1
Проверка        : vsftpd-3.0.5-6.el9.x86_64 1/1
Установлен:
vsftpd-3.0.5-6.el9.x86_64
Выполнено!
```

Рис. 2.1: 1

Далее запускаем службу Very Secure FTP и проверяем статус службы Very Secure FTP (рис. 2.2).

```

[root@localhost ~]# systemctl start vsftpd
[root@localhost ~]# systemctl status vsftpd
* vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
   Active: active (running) since Fri 2025-09-26 13:16:18 +03; 14s ago
     Process: 4400 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exited, status=0/SUCCESS)
    Main PID: 4401 (vsftpd)
       Tasks: 1 (limit: 22974)
      Memory: 736.0K
         CPU: 7ms
        CGroup: /system.slice/vsftpd.service
               └─4401 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

сен 26 13:16:18 localhost.localdomain systemd[1]: Starting Vsftpd ftp daemon...
сен 26 13:16:18 localhost.localdomain systemd[1]: Started Vsftpd ftp daemon.

```

Рис. 2.2: 2

После чего чтобы добавить службу Very Secure FTP в автозапуск при загрузке операционной системы используем команду `systemctl enable vsftpd` затем проверим статус службы после чего удаляем службу из автозапуска используя команду `systemctl disable` и снова проверяем статус (рис.2.3).

```

[root@localhost ~]# systemctl enable vsftpd
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /usr/lib/systemd/system/vsftpd.service.
[root@localhost ~]# systemctl status vsftpd
* vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
   Active: active (running) since Fri 2025-09-26 13:16:18 +03; 1min 55s ago
     Process: 4401 (vsftpd)
       Tasks: 1 (limit: 22974)
      Memory: 736.0K
         CPU: 7ms
        CGroup: /system.slice/vsftpd.service
               └─4401 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

сен 26 13:16:18 localhost.localdomain systemd[1]: Starting Vsftpd ftp daemon...
сен 26 13:16:18 localhost.localdomain systemd[1]: Started Vsftpd ftp daemon.
[root@localhost ~]# systemctl disable vsftpd
Removed "/etc/systemd/system/multi-user.target.wants/vsftpd.service".
[root@localhost ~]# systemctl status vsftpd
* vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
   Active: active (running) since Fri 2025-09-26 13:16:18 +03; 2min 19s ago
     Process: 4401 (vsftpd)
       Tasks: 1 (limit: 22974)
      Memory: 736.0K
         CPU: 7ms
        CGroup: /system.slice/vsftpd.service
               └─4401 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

сен 26 13:16:18 localhost.localdomain systemd[1]: Starting Vsftpd ftp daemon...
сен 26 13:16:18 localhost.localdomain systemd[1]: Started Vsftpd ftp daemon.

```

Рис. 2.3: 3

Далее выводим символические ссылки ответственные за запуск различных сервисов после первого вывода мы увидим что ссылки на `vsftpd.service` не существует. После чего добавим службу Very Secure FTP в автозапуск и еще раз выведем ссылки и обнаружим что `vsftpd.service` добавилась (рис. 2.4).

```

[root@localhost ~]# ls /etc/systemd/system/multi-user.target.wants
atd.service      cups.service      libstoragemgmt.service  remote-fs.target  tuned.service
auditd.service  cups.service      mcelog.service          rsyslog.service  vboxadd.service
avahi-daemon.service  firewallld.service  mdmonitord.service      smartd.service   vboxadd-service.service
chronyd.service  irqbalance.service  ModemManager.service    sshd.service     vmtoolsd.service
crond.service    kdump.service      NetworkManager.service  sssd.service

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

```

Рис. 2.4: 4

Затем проверяем снова статус службы и увидим что файл юнита состояния изменено disabled на enabled(рис. 2.5).

```
[root@localhost ~]# systemctl status vsftpd
vsftpd.service - Vsftpd ftp daemon
Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
Active: active (running) since Fri 2025-09-26 13:16:18 +03; 4min 22s ago
Main PID: 4401 (vsftpd)
Tasks: 1 (limit: 22974)
Memory: 736.0K
CPU: 7ms
CGroup: /system.slice/vsftpd.service
        └─4401 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

сеп 26 13:16:18 localhost.localdomain systemd[1]: Starting Vsftpd ftp daemon...
сеп 26 13:16:18 localhost.localdomain systemd[1]: Started Vsftpd ftp daemon.
```

Рис. 2.5: 5

Теперь выведем список зависимостей юнита (рис. 2.6).

```
[root@localhost ~]# systemctl list-dependencies vsftpd
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
├─systemd-binfmt.service
├─systemd-boot-random-seed.service
├─systemd-boot-update.service
├─systemd-firstboot.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-ocrmachine.service
```

Рис. 2.6: 6

Затем выведем список юнитов которые зависят от данного юнита(рис. 2.7).

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

Рис. 2.7: 7

После чего мы устанавливаем iptables (рис. 2.8).



```
[root@localhost ~]# dnf -y install iptables*
Последняя проверка окончания срока действия метаданных: 0:06:32 назад, Пт 26 сен 2025 13:16:03.
Пакет iptables-libs-1.8.10-11.el9_5.x86_64 уже установлен.
Пакет iptables-nft-1.8.10-11.el9_5.x86_64 уже установлен.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий  Размер
=====
Установка:
iptables-devel      x86_64      1.8.10-11.el9_5      appstream    16 k
iptables-legacy     x86_64      1.8.10-11.1.el9      epel         50 k
iptables-legacy-devel x86_64      1.8.10-11.1.el9      epel         14 k
iptables-legacy-libs x86_64      1.8.10-11.1.el9      epel         38 k
iptables-nft-services noarch      1.8.10-11.el9_5      appstream    19 k
iptables-services   noarch      1.8.10-11.1.el9      epel         17 k
iptables-utils      x86_64      1.8.10-11.el9_5      baseos       41 k
=====
Результат транзакции
=====
Установка 7 Пакетов
Объем загрузки: 195 k
Объем изменений: 365 k
Загрузка пакетов:
=====
[===] --- B/s | 0 B --- ETA
```

Рис. 2.8: 8

Проверяем статус firewalld и iptables (рис. 2.9).

```
[root@localhost ~]# 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 2025-09-26 13:07:26 +03; 15min ago
     Docs: man:firewalld(1)
   Main PID: 987 (firewalld)
    Tasks: 2 (limit: 22974)
   Memory: 36.8M
      CPU: 1.210s
   CGroup: /system.slice/firewalld.service
           └─987 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid

сен 26 13:07:25 localhost systemd[1]: Starting firewalld - dynamic firewall daemon...
сен 26 13:07:26 localhost systemd[1]: Started firewalld - dynamic firewall daemon.
[root@localhost ~]# systemctl status iptables
● iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; preset: disabled)
   Active: inactive (dead)
```

Рис. 2.9: 9

Пробуем запустить firewalld и iptables (рис. 2.10).

```
[root@localhost ~]# systemctl start firewalld
[root@localhost ~]# systemctl start iptables
[root@localhost ~]# status iptables
bash: status: команда не найдена...
[root@localhost ~]# systemctl status iptables
● iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; preset: disabled)
   Active: active (exited) since Fri 2025-09-26 13:23:48 +03; 1min 15s ago
   Process: 5135 ExecStart=/usr/libexec/iptables/iptables.init start (code=exited, status=0/SUCCESS)
   Main PID: 5135 (code=exited, status=0/SUCCESS)
      CPU: 86ms

сен 26 13:23:48 localhost.localdomain systemd[1]: Starting IPv4 firewall with iptables...
сен 26 13:23:48 localhost.localdomain iptables.init[5135]: iptables: Applying firewall rules: [ OK ]
сен 26 13:23:48 localhost.localdomain systemd[1]: Finished IPv4 firewall with iptables.
[root@localhost ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
   Active: inactive (dead) since Fri 2025-09-26 13:23:48 +03; 1min 34s ago
  Duration: 16min 21.950s
     Docs: man:firewalld(1)
   Process: 987 ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS (code=exited, status=0/SUCCESS)
   Main PID: 987 (code=exited, status=0/SUCCESS)
      CPU: 1.294s

сен 26 13:07:25 localhost systemd[1]: Starting firewalld - dynamic firewall daemon...
сен 26 13:07:26 localhost systemd[1]: Started firewalld - dynamic firewall daemon.
сен 26 13:23:48 localhost.localdomain systemd[1]: Stopping firewalld - dynamic firewall daemon...
сен 26 13:23:48 localhost.localdomain systemd[1]: firewalld.service: Deactivated successfully.
сен 26 13:23:48 localhost.localdomain systemd[1]: Stopped firewalld - dynamic firewall daemon.
сен 26 13:23:48 localhost.localdomain systemd[1]: firewalld.service: Consumed 1.294s CPU time.
```

Рис. 2.10: 10

После чего выведем на экран настройки конфликтов для firewalld и ока-

жется что у него конфликт будет называться iptables.service iptable.service ebtables.service ipset.service (рис. 2.11).

```
[root@localhost ~]# 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 iptables.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

[Install]
WantedBy=multi-user.target
Alias=dbus-org.fedoraproject.FirewallD1.service
```

Рис. 2.11: 11

Теперь выведем для iptables и там не будет конфликтов (рис. 2.12).

```
[root@localhost ~]# 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
```

Рис. 2.12: 12

После чего выгружаем службу iptables и загружаем службу firewalld и блокируем запуск iptables(рис. 2.13).

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

Рис. 2.13: 13

После блокировки iptables пробуем запустить его и у нас появляется ошибка указывающая что служба замаскирована и по этой причине не может быть запущена.Пробуем добавить iptables в автозапуск и убеждаемся что сервис будет не активен и также будет отображаться как замаскированным (рис. [-fig. 2.14).

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

Рис. 2.14: 14

После чего переходим в каталог system и находим список всех целей которые можно изолировать (рис. 2.15).

```
[root@localhost 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@localhost system]#
```

Рис. 2.15: 15

Далее переключаем операционную систему в режим восстановления (рис. 2.16).

```
[sikaznacheev@localhost ~]$ su -
Пароль:
[root@localhost ~]# cd /usr/lib/systemd/system
[root@localhost system]# systemctl isolate rescue.target
```

Рис. 2.16: 16

```
You are in rescue mode. After logging in, type "journalctl -xb" to view
system logs, "systemctl reboot" to reboot, "systemctl default" or "exit"
to boot into default mode.
Для продолжения введите пароль root
(или нажмите Control-D для продолжения):
[root@localhost ~]# systemctl isolate reboot.target
```

Рис. 2.17: 17

После перезапуска системы переходим в супер пользователя root и выводим установленную цель по умолчанию (рис. 2.18).

```
[s1kaznacheev@localhost ~]$ su -  
Пароль:  
[root@localhost ~]# systemctl get-default  
graphical.target
```

Рис. 2.18: 18

Далее устанавливаем цель используем команду `systemctl set-default multi-user.target` и перезагружаем систему командой `reboot`. (рис. 2.19).

```
[root@localhost ~]# 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.
```

Рис. 2.19: 19

Затем проверяем что система загрузилась в текстовом режиме. После чего получаем полномочия администратора для и вводим команду `systemctl set-default graphical.target` и вновь перезагружаем командой `reboot`. Убеждаемся что загрузилась в графическом режиме (рис. 2.20).

```
Rocky Linux 9.6 (Blue Onyx)  
Kernel 5.14.0-578.37.1.el9_6.x86_64 on x86_64  
  
Activate the web console with: systemctl enable --now cockpit.socket  
  
localhost login: root  
Password:  
Last login: Fri Sep 26 13:38:47 on pts/0  
(root@localhost ~) # 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@localhost ~) # reboot
```

Рис. 2.20: 20

### 3 Контрольные вопросы

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

Ответ - это объект systemd описывающий службу, устройство точку монтирования и т.д. Пример - service, target, mount

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

Ответ - чтобы проверить что цель больше не включена в автозапуск надо ввести команду `systemctl is-enabled`

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

Ответ - чтобы показать все загруженные сервисные юниты надо ввести команду `systemctl list-units --type=service`

4. Как создать отребность (wants) в сервисе?

Ответ - чтобы создать отребность (wants) в сервисе нужна команда `systemctl add-wants .target .service`

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

Ответ - чтобы переключить текущее нужно переключиться в режим восстановления благодаря команде `systemctl isolate rescue.target`

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

Ответ - Цель не может быть изолирована если она не установлена как AllowIsolate=yes в unit-файле

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?

Ответ - Показать зависимость от службы командой `systemctl list-dependencies`

## 4 Выводы

В результате выполнения лабораторной работы я получил навыки работы управления системными службами операционной системе посредством systemd