

# **Лабораторная работа №5**

**Отчет**

Славинский Владислав Вадимович

# 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>Выводы</b>	<b>17</b>
<b>4</b>	<b>Ответы на контрольные вопросы</b>	<b>18</b>

# Список иллюстраций

2.1	Переход в режим суперпользователя . . . . .	6
2.2	Проверка статуса службы . . . . .	6
2.3	Установка vsftpd . . . . .	7
2.4	Запуск vsftpd . . . . .	7
2.5	Проверка статуса службы . . . . .	7
2.6	Добавление в автозапуск . . . . .	8
2.7	Отключение службы . . . . .	8
2.8	Вывод символических ссылок . . . . .	9
2.9	Вывод символических ссылок после добавления vsftpd в автозапуск . . . . .	9
2.10	Проверка статуса службы . . . . .	9
2.11	Вывод списка зависимостей юнита . . . . .	9
2.12	Вывод списка юнитов, которые зависят от данного типа . . . . .	10
2.13	Установка iptables . . . . .	10
2.14	Проверка статуса firewalld . . . . .	11
2.15	Проверка статуса iptables . . . . .	11
2.16	Запуск firewalld и iptables . . . . .	11
2.17	Проверка статуса firewalld . . . . .	11
2.18	Проверка статуса iptables . . . . .	12
2.19	Ввод команды для анализа ошибок . . . . .	12
2.20	Ввод команды для анализа ошибок . . . . .	13
2.21	Выгрузка iptables и загрузка firewalld . . . . .	13
2.22	Блокировка запуска iptables . . . . .	13
2.23	Запуск iptables . . . . .	13
2.24	Попытка добавления iptables в автозапуск . . . . .	14
2.25	Список целей, которые можно изолировать . . . . .	14
2.26	Переключение операционной системы в режим восстановления . . . . .	14
2.27	Перезапуск операционной системы с изменениями . . . . .	15
2.28	Вывод цели по умолчанию . . . . .	15
2.29	Запуск текстового режима . . . . .	15
2.30	Возвращение на графический режим . . . . .	16
2.31	Запуск в графическом режиме . . . . .	16

## **Список таблиц**

# 1 Цель работы

Получить навыки управления системными службами операционной системы посредством systemd.

## 2 Выполнение лабораторной работы

В консоли перейдем в режим работы суперпользователя, используя команду `su -`. (рис. 2.1)

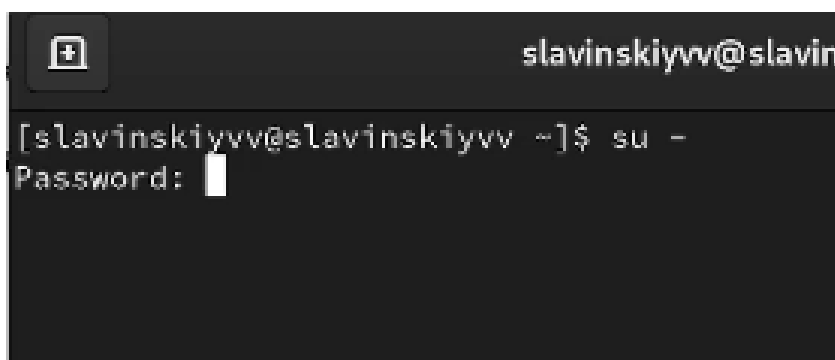
A terminal window with a dark background. The title bar shows a window icon and the text 'slavinskiyv@slavin'. The terminal content shows a user prompt '[slavinskiyv@slavinskiyv ~]\$' followed by the command 'su -'. Below this, the prompt 'Password:' is shown with a small white cursor character.

Рис. 2.1: Переход в режим суперпользователя

Проверим статус службы Very Secure FTP с помощью команды `systemctl status vsftpd`. Служба отключена, так как она не установлена. (рис. 2.2)

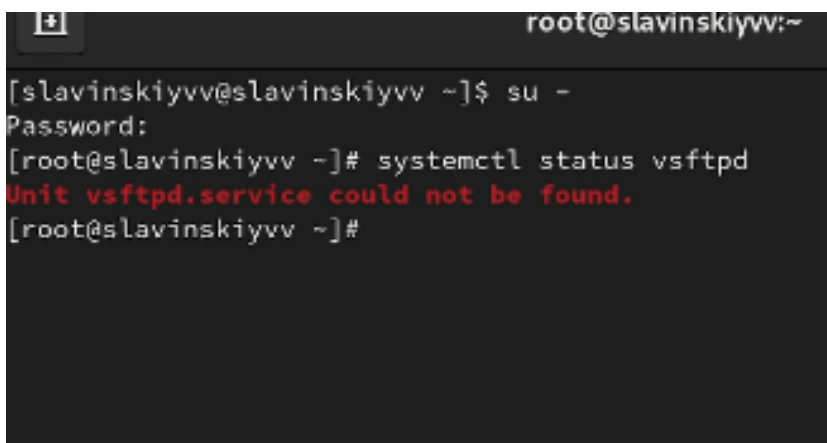
A terminal window with a dark background. The title bar shows a window icon and the text 'root@slavinskiyv:~'. The terminal content shows the user prompt '[slavinskiyv@slavinskiyv ~]\$' followed by the command 'su -'. Below this, the prompt 'Password:' is shown. Then, the root prompt '[root@slavinskiyv ~]#' is shown, followed by the command 'systemctl status vsftpd'. The output of the command is 'Unit vsftpd.service could not be found.', which is displayed in red text.

Рис. 2.2: Проверка статуса службы

Установим службу Very Secure FTP: `dnf -y install vsftpd`. (рис. 2.3)

```
Unit vsftpd.service could not be found.  
[root@slavinskiyv ~]# dnf -y install vsftpd
```

Рис. 2.3: Установка vsftp

Запустим службу Very Secure FTP: `systemctl start vsftpd`. (рис. 2.4)

```
Complete!  
[root@slavinskiyv ~]# systemctl start vsftpd  
[root@slavinskiyv ~]#
```

Рис. 2.4: Запуск vsftp

Проверим статус службы Very Secure FTP с помощью команды `systemctl status vsftpd`. (рис. 2.5)

```
Installed:  
vsftpd-3.0.5-6.el9.x86_64  
  
Complete!  
[root@slavinskiyv ~]# systemctl start vsftpd  
[root@slavinskiyv ~]# systemctl status vsftpd  
● vsftpd.service - Vsftpd ftp daemon  
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: enabled)  
   Active: active (running) since Fri 2025-10-03 18:48:17 MSK; 8s ago  
 Process: 7240 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exited, status=0/SUCCESS)  
 Main PID: 7241 (vsftpd)  
    Tasks: 1 (limit: 48747)  
  Memory: 740.0K  
     CPU: 2ms  
   CGroup: /system.slice/vsftpd.service  
           └─7241 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf  
  
Oct 03 18:48:17 slavinskiyv systemd[1]: Starting Vsftpd ftp daemon...  
Oct 03 18:48:17 slavinskiyv systemd[1]: Started Vsftpd ftp daemon.  
lines 1-13/13 (END)
```

Рис. 2.5: Проверка статуса службы

У нас служба работает, но у нас она не будет работать при автоматическом запуске операционной системы, давайте её добавим в автоматический запуск

с помощью команды `systemctl enable vsftpd`. И как видим, служба добавилась в автозапуск. (рис. 2.6)

```
[1]+  Stopped                  systemctl status vsftpd
[root@slavinskiyv ~]# systemctl enable vsftpd
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /usr/lib/systemd/system/vsftpd.service.
[root@slavinskiyv ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
   Active: active (running) since Fri 2025-10-03 18:48:17 MSK; 1min 36s ago
     Main PID: 7241 (vsftpd)
        Tasks: 1 (limit: 48747)
      Memory: 740.0K
         CPU: 2ms
    CGroup: /system.slice/vsftpd.service
            └─7241 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Oct 03 18:48:17 slavinskiyv systemd[1]: Starting Vsftpd ftp daemon...
Oct 03 18:48:17 slavinskiyv systemd[1]: Started Vsftpd ftp daemon.
[root@slavinskiyv ~]#
```

Рис. 2.6: Добавление в автозапуск

Теперь удалим службу из автозапуска через команду `systemctl disable vsftpd`.  
Теперь служба удалилась из автозапуска. (рис. 2.7)

```
Tasks: 1 (limit: 48747)
Memory: 740.0K
   CPU: 2ms
CGroup: /system.slice/vsftpd.service
        └─7241 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Oct 03 18:48:17 slavinskiyv systemd[1]: Starting Vsftpd ftp daemon...
Oct 03 18:48:17 slavinskiyv systemd[1]: Started Vsftpd ftp daemon.
[root@slavinskiyv ~]# systemctl disable vsftpd
Removed "/etc/systemd/system/multi-user.target.wants/vsftpd.service".
[root@slavinskiyv ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
   Active: active (running) since Fri 2025-10-03 18:48:17 MSK; 2min 3s ago
     Main PID: 7241 (vsftpd)
        Tasks: 1 (limit: 48747)
      Memory: 740.0K
         CPU: 2ms
    CGroup: /system.slice/vsftpd.service
            └─7241 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Oct 03 18:48:17 slavinskiyv systemd[1]: Starting Vsftpd ftp daemon...
Oct 03 18:48:17 slavinskiyv systemd[1]: Started Vsftpd ftp daemon.
[root@slavinskiyv ~]#
```

Рис. 2.7: Отключение службы

Выведем на экран символические ссылки, ответственные за запуск различных сервисов: `ls /etc/systemd/system/multi-user.target.wants`. В данном случае мы не видим `vsftpd.service`. (рис. 2.8)



```

Oct 03 18:48:17 slavinskiyvv systemd[1]: Started Vsftpd ftp daemon.
[root@slavinskiyvv ~]# ls /etc/systemd/system/multi-user.target.wants
audit.service      cups.path          mcelog.service     rsyslog.service
auditd.service     cups.service       mdmonitor.service  smartd.service
avahi-daemon.service  irqbalance.service  ModemManager.service  sshd.service
chronyd.service      kdump.service       NetworkManager.service  sssd.service
crond.service        libstoragemgmt.service  remote-fs.target     tuned.service
[root@slavinskiyvv ~]#

```

Рис. 2.8: Вывод символических ссылок

Теперь снова добавим vsftpd в автозапуск и проверим, появился ли vsftpd.service. Как видим, у нас vsftpd появился. (рис. 2.9)

```

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

```

Рис. 2.9: Вывод символических ссылок после добавления vsftpd в автозапуск

Снова проверим статус службы Very Secure FTP. У нас служба будет включена после перезапуска системы. (рис. 2.10)

```

crond.service      crond.service      remote-fs.target    rsyslog.service
[root@slavinskiyvv ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
   Active: active (running) since Fri 2025-10-03 18:48:17 MSK; 3min 31s ago
     Main PID: 7241 (vsftpd)
        Tasks: 1 (limit: 48747)
       Memory: 740.0K
          CPU: 2ms
      CGroup: /system.slice/vsftpd.service
              └─7241 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Oct 03 18:48:17 slavinskiyvv systemd[1]: Starting Vsftpd ftp daemon...
Oct 03 18:48:17 slavinskiyvv systemd[1]: Started Vsftpd ftp daemon.
[root@slavinskiyvv ~]#

```

Рис. 2.10: Проверка статуса службы

Выведем на экран список зависимостей юнита: systemctl list-dependencies vsftpd. (рис. 2.11)

```

Oct 03 18:48:17 slavinskiyvv systemd[1]: Starting Vsftpd ftp daemon...
Oct 03 18:48:17 slavinskiyvv systemd[1]: Started Vsftpd ftp daemon.
[root@slavinskiyvv ~]# 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
└─...

```

Рис. 2.11: Вывод списка зависимостей юнита

Выведем на экран список юнитов, которые зависят от данного юнита: `systemctl list-dependencies vsftpd --reverse`. (рис. 2.12)

```
lines 3-30/58 51%
[2]+  Stopped                  systemctl list-dependencies vsftpd
[root@slavinskiyvv ~]# systemctl list-dependencies vsftpd --reverse
vsftpd.service
├─multi-user.target
└─graphical.target
[root@slavinskiyvv ~]#
```

Рис. 2.12: Вывод списка юнитов, которые зависят от данного типа

Дальше установим iptables: `dnf -y install iptables*`. (рис. 2.13)

```
[root@slavinskiyvv ~]# dnf -y install iptables*
Last metadata expiration check: 1:25:26 ago on Fri 03 Oct 2025 05:27:59 |
Dependencies resolved.
=====
Package                                Architecture      Version
=====
Installing:
  iptables-devel                        x86_64            1.8.10-11.el9_5
  iptables-legacy                      x86_64            1.8.10-11.1.el9
  iptables-legacy-devel                x86_64            1.8.10-11.1.el9
  iptables-legacy-libs                 x86_64            1.8.10-11.1.el9
  iptables-libs                        x86_64            1.8.10-11.el9_5
  iptables-nft                         x86_64            1.8.10-11.el9_5
  iptables-nft-services                noarch            1.8.10-11.el9_5
  iptables-services                    noarch            1.8.10-11.1.el9
  iptables-utils                       x86_64            1.8.10-11.el9_5
Installing dependencies:
  libnetfilter_conntrack                x86_64            1.0.9-1.el9
  libnftnl                              x86_64            1.0.1-23.el9_5
  libnftnl                              x86_64            1.2.6-4.el9_4
Transaction Summary
=====
Install 12 Packages

Total download size: 952 k
Installed size: 3.1 M
Downloading Packages:
[=====
```

Рис. 2.13: Установка iptables

Проверим статус firewalld: `systemctl status firewalld`. (рис. 2.14)

```
[root@slavinskiyvv ~]# 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 2025-10-03 18:58:44 MSK; 3min 15s ago
     Docs: man:firewalld(1)
    Main PID: 8532 (firewalld)
      Tasks: 2 (limit: 48747)
     Memory: 26.4M
        CPU: 318ms
    CGroup: /system.slice/firewalld.service
            └─8532 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid

Oct 03 18:58:44 slavinskiyvv systemd[1]: Starting firewalld - dynamic firewall daemon...
Oct 03 18:58:44 slavinskiyvv systemd[1]: Started firewalld - dynamic firewall daemon.
[root@slavinskiyvv ~]#
```

Рис. 2.14: Проверка статуса firewalld

Проверим статус iptables: `systemctl status iptables`. Здесь мы видим, что служба инактивна, и не запущена а автозапуске(рис. 2.15)

```
Oct 03 18:58:44 slavinskiyvv systemd[1]: Starting firewalld - dynamic firewall daemon...
Oct 03 18:58:44 slavinskiyvv systemd[1]: Started firewalld - dynamic firewall daemon.
[root@slavinskiyvv ~]# systemctl status iptables
○ iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; preset: disabled)
   Active: inactive (dead)
```

Рис. 2.15: Проверка статуса iptables

Попробуем запустить firewalld и iptables: `systemctl start firewalld,systemctl start iptables`.(рис. 2.16)

```
lines 1-12/12 (END)
[6] Stopped                                systemctl status iptables
[root@slavinskiyvv ~]# systemctl start firewalld
[root@slavinskiyvv ~]# systemctl start iptables
[root@slavinskiyvv ~]#
и загрузите службу firewalld
```

Рис. 2.16: Запуск firewalld и iptables

Посмотрим статус firewalld. firewalld у нас теперь не запущена.(рис. 2.17)

```
[root@slavinskiyvv ~]# systemctl start firewalld
[root@slavinskiyvv ~]# systemctl start iptables
[root@slavinskiyvv ~]# 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 2025-10-03 19:03:27 MSK; 8s ago
     Duration: 4min 42.877s
     Docs: man:firewalld(1)
    Process: 8532 ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS (code=exited, status=0/SUCCESS)
   Main PID: 8532 (code=exited, status=0/SUCCESS)
        CPU: 352ms
```

Рис. 2.17: Проверка статуса firewalld

Посмотрим статус iptables. И тут уже понятно, что одна служба деактивируется, а другая включается, поскольку iptables запустилась.(рис. 2.18)

```
Oct 03 19:03:27 slavinskiyvv systemd[1]: Stopped firewalld - dynamic firewall daemon.
lines 1-14/14 (END)
[7]# systemctl status firewalld
[root@slavinskiyvv ~]# 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 2025-10-03 19:03:27 MSK; 22s ago
     Process: 8686 ExecStart=/usr/libexec/iptables/iptables.init start (code=exited, status=0/SUCCESS)
    Main PID: 8686 (code=exited, status=0/SUCCESS)
      CPU: 11ms

Oct 03 19:03:27 slavinskiyvv systemd[1]: Starting IPv4 firewall with iptables...
Oct 03 19:03:27 slavinskiyvv iptables.init[8686]: iptables: Applying firewall rules: [ OK ]
Oct 03 19:03:27 slavinskiyvv systemd[1]: Finished IPv4 firewall with iptables.
[root@slavinskiyvv ~]#
```

Рис. 2.18: Проверка статуса iptables

Введем cat /usr/lib/systemd/system/firewalld.service, чтобы посмотреть ошибки. И вот мы видим, с чем конфликтует служба firewalld.(рис. 2.19)

```
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 --noupdate $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@slavinskiyvv ~]#
```

Рис. 2.19: Ввод команды для анализа ошибок

Введем то же самое, только для iptables: cat /usr/lib/systemd/system/iptables.service. Но тут, мы не видим никаких ошибок.(рис. 2.20)

```

Alias=dbus-org.fedoraproject.FirewallD1.service
[root@slavinskiyvv ~]# 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@slavinskiyvv ~]#

```

Рис. 2.20: Ввод команды для анализа ошибок

Выгрузим службу iptables (на всякий случай, чтобы убедиться, что данная служба не загружена в систему): `systemctl stop iptables`, и загрузим службу firewalld `systemctl start firewalld`. (рис. 2.21)

```

wantedBy=multi-user.target
[root@slavinskiyvv ~]# systemctl stop iptables
[root@slavinskiyvv ~]# systemctl start firewalld
[root@slavinskiyvv ~]#

```

Рис. 2.21: Выгрузка iptables и загрузка firewalld

Заблокируем запуск iptables, введя команду `systemctl mask iptables`. (рис. 2.22)

```

[root@slavinskiyvv ~]# systemctl start firewalld
[root@slavinskiyvv ~]# systemctl mask iptables
Created symlink /etc/systemd/system/iptables.service → /dev/null.
[root@slavinskiyvv ~]#

```

Рис. 2.22: Блокировка запуска iptables

Теперь попробуем запустить iptables. Видим, что у нас ошибка, так как мы эту службу замаскировали. (рис. 2.23)

```

[root@slavinskiyvv ~]# systemctl start firewalld
[root@slavinskiyvv ~]# systemctl mask iptables
Created symlink /etc/systemd/system/iptables.service → /dev/null.
[root@slavinskiyvv ~]# systemctl start iptables
Failed to start iptables.service: Unit iptables.service is masked.
[root@slavinskiyvv ~]#

```

Рис. 2.23: Запуск iptables

Попробуем добавить iptables в автозапуск, но сервис будет неактивен, а статус загрузки отобразился как замаскированный.(рис. 2.24)

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

Рис. 2.24: Попытка добавления iptables в автозапуск

Дальше перейдем каталог systemd и найдем список всех целей, которые можно изолировать: `cd /usr/lib/systemd/system, grep Isolate *.target`.(рис. 2.25)

```
[8]+  Stopped                               systemctl status iptables
[root@slavinskiyvv ~]# cd /usr/lib/systemd/system
[root@slavinskiyvv 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@slavinskiyvv system]#
```

Рис. 2.25: Список целей, которые можно изолировать

Переключим операционную систему в режим восстановления: `systemctl isolate rescue.target`.(рис. 2.26)

```
runlevel4.target:AllowIsolate=yes
runlevel5.target:AllowIsolate=yes
runlevel6.target:AllowIsolate=yes
system-update.target:AllowIsolate=yes
[root@slavinskiyvv system]# systemctl isolate rescue.target
```

Рис. 2.26: Переключение операционной системы в режим восстановления

Перезапустим операционную систему следующим образом: `systemctl isolate reboot.target`.(рис. 2.27)

```
system logs; "systemctl reboot" to reboot, "systemctl default" or "exit"
to boot into default mode.
Give root password for maintenance
(or press Control-D to continue):
[root@slavinskiy ~]# systemctl isolate reboot.target
```

Рис. 2.27: Перезапуск операционной системы с изменениями

Теперь вводим команду `systemctl get-default`, чтобы узнать установленную по умолчанию цель. Видим, что запускается система по умолчанию в графическом режиме.(рис. 2.28)

```
[root@slavinskiy ~]# cat -
Password:
[root@slavinskiy ~]# systemctl get-default
graphical.target
[root@slavinskiy ~]#
```

Рис. 2.28: Вывод цели по умолчанию

Для запуска по умолчанию текстового режима введём `systemctl set-default multi-user.target` и перезагружаем.(рис. 2.29)

```
graphical.target
[root@slavinskiy ~]# systemctl set-default
Too few arguments.
[root@slavinskiy ~]# 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@slavinskiy ~]#
```

Рис. 2.29: Запуск текстового режима

Чтобы нам обратно вернуться в графический режим, нужно перейти на root и ввести команду `systemctl set-default graphical.target`.(рис. 2.30)

```
root@kali:~# systemctl set-default graphical.target
root@kali:~# systemctl set-default graphical.target
Created symlink /etc/systemd/system/default.target → /usr/lib/systemd/system/graphical.target.
```

Рис. 2.30: Возвращение на графический режим

Перезагружаем и видим, мы снова в графическом режиме.(рис. 2.31)

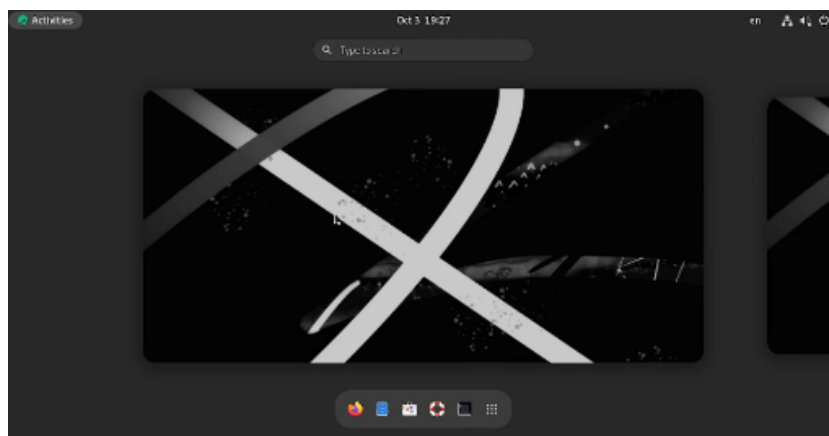


Рис. 2.31: Запуск в графическом режиме



## **3 Выводы**

В ходе выполнения лабораторной работы были получены навыки управления системными службами операционной системы посредством systemd.

## 4 Ответы на контрольные вопросы

1. Юнит- это файл конфигурации с определенным синтаксисом, который описывает системную службу. Примеры: `.service`, `.target`, `.mount`.
2. `systemctl is-enabled`
3. `systemctl list-units --type=service`
4. `systemctl enable` (название юнита)
5. `systemctl isolate rescue.target`
6. Цель не может быть изолирована, если в её файле конфигурации `.target` отсутствует или имеет значение по директива `AllowIsolate`.
7. `systemctl list-dependencies (служба) --reverse`