

Лабораторная работа №5

Основы администрирования операционных систем

Верниковская Е. А., НПИбд-01-23

28 сентября 2024

Российский университет дружбы народов, Москва, Россия

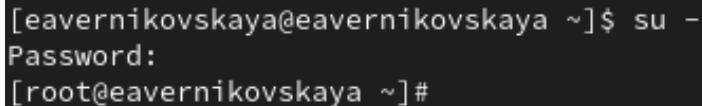
Вводная часть

Получить навыки управления системными службами операционной системы посредством `systemd`.

1. Выполнить основные операции по запуску (останову), определению статуса, добавлению (удалению) в автозапуск и пр. службы Very Secure FTP.
2. Продемонстрировать навыки по разрешению конфликтов юнитов для служб `firewalld` и `iptables`.
3. Продемонстрировать навыки работы с изолированными целями.

Выполнение лабораторной работы

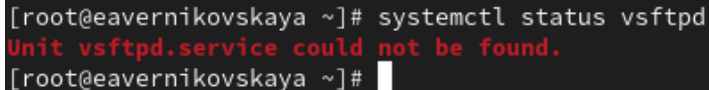
В консоли переходим в режим работы суперпользователя, используя *su* - (рис. 1)



```
[eavernikovskaya@eavernikovskaya ~]$ su -  
Password:  
[root@eavernikovskaya ~]#
```

Рис. 1: Режим суперпользователя

Проверяем статус службы Very Secure FTP: `systemctl status vsftpd`. Вывод команды показывает, что сервис в настоящее время отключён, так как служба Very Secure FTP не установлена (рис. 2)



```
[root@eavernikovskaya ~]# systemctl status vsftpd
Unit vsftpd.service could not be found.
[root@eavernikovskaya ~]#
```

Рис. 2: Статус службы Very Secure FTP (1)

Устанавливаем службу Very Secure FTP: `dnf -y install vsftpd` (рис. 3)

```
[root@eavernikovskaya ~]# dnf -y install vsftpd
Rocky Linux 9 - BaseOS                               8.8 kB/s | 4.1 kB   00:00
Rocky Linux 9 - BaseOS                               1.8 MB/s | 2.3 MB   00:01
Rocky Linux 9 - AppStream                             11 kB/s | 4.5 kB   00:00
Rocky Linux 9 - AppStream                             3.0 MB/s | 8.0 MB   00:02
Rocky Linux 9 - Extras                                6.9 kB/s | 2.9 kB   00:00
Dependencies resolved.
=====
Package            Architecture    Version           Repository        Size
=====
Installing:
vsftpd             x86_64         3.0.5-5.el9       appstream         157 k
=====
Transaction Summary
=====
Install 1 Package

Total download size: 157 k
Installed size: 347 k
Downloading Packages:
vsftpd-3.0.5-5.el9.x86_64.rpm                        355 kB/s | 157 kB   00:00
-----
Total                                                    205 kB/s | 157 kB   00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                1/1
  Installing     : vsftpd-3.0.5-5.el9.x86_64      1/1
  Running scriptlet: vsftpd-3.0.5-5.el9.x86_64      1/1
  Verifying      : vsftpd-3.0.5-5.el9.x86_64      1/1

Installed:
vsftpd-3.0.5-5.el9.x86_64

Complete!
[root@eavernikovskaya ~]#
```

Рис. 3: Установка службы Very Secure FTP

Запускаем службу Very Secure FTP:: *systemctl start vsftpd* (рис. 4)

A terminal window with a black background and white text. The prompt is [root@eavernikovskaya ~]#. The command systemctl start vsftpd has been entered. The prompt is repeated on the next line with a white cursor block at the end.

```
[root@eavernikovskaya ~]# systemctl start vsftpd
[root@eavernikovskaya ~]#
```

Рис. 4: Запуск службы Very Secure FTP

Снова проверяем статус службы Very Secure FTP. Вывод команды показывает, что служба в настоящее время работает, но не будет активирована при перезапуске операционной системы (рис. 5)

```
[root@eavernikovskaya ~]# systemctl status vsftpd
• vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
   Active: active (running) since Sat 2024-09-28 14:15:40 MSK; 20s ago
     Process: 2740 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exited, status=0/SUCCESS)
    Main PID: 2741 (vsftpd)
       Tasks: 1 (limit: 12205)
      Memory: 708.0K
         CPU: 4ms
       CGroup: /system.slice/vsftpd.service
               └─2741 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 28 14:15:40 eavernikovskaya.localdomain systemd[1]: Starting Vsftpd ftp daemon...
Sep 28 14:15:40 eavernikovskaya.localdomain systemd[1]: Started Vsftpd ftp daemon.
[root@eavernikovskaya ~]#
```

Рис. 5: Статус службы Very Secure FTP после установки

Добавляем службу Very Secure FTP в автозапуск при загрузке операционной системы, используя команду *systemctl enable vsftpd* (рис. 6)

```
[root@eavernikovskaya ~]# systemctl enable vsftpd
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /usr/lib/systemd/system/vsftpd.service.
[root@eavernikovskaya ~]#
```

Рис. 6: Добавление службы Very Secure FTP в автозапуск (1)

Проверяем статус службы Very Secure FTP (рис. 7)

```
[root@eavernikovskaya ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
   Active: active (running) since Sat 2024-09-28 14:15:40 MSK; 2min 30s ago
     Main PID: 2741 (vsftpd)
        Tasks: 1 (limit: 12205)
       Memory: 708.0K
          CPU: 4ms
       CGroup: /system.slice/vsftpd.service
               └─2741 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 28 14:15:40 eavernikovskaya.localdomain systemd[1]: Starting Vsftpd ftp daemon...
Sep 28 14:15:40 eavernikovskaya.localdomain systemd[1]: Started Vsftpd ftp daemon.
[root@eavernikovskaya ~]#
```

Рис. 7: Статус службы Very Secure FTP после добавления в автозапуск

Удаляем службу из автозапуска, используя команду *systemctl disable vsftpd*, и снова проверяем её статус (рис. 8), (рис. 9)

```
[root@eavernikovskaya ~]# systemctl disable vsftpd
Removed "/etc/systemd/system/multi-user.target.wants/vsftpd.service".
[root@eavernikovskaya ~]#
```

Рис. 8: Удаление службы Very Secure FTP из автозапуска

```
[root@eavernikovskaya ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
   Active: active (running) since Sat 2024-09-28 14:15:40 MSK; 3min 28s ago
     Main PID: 2741 (vsftpd)
        Tasks: 1 (limit: 12205)
       Memory: 708.0K
          CPU: 4ms
       CGroup: /system.slice/vsftpd.service
               └─2741 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 28 14:15:40 eavernikovskaya.localdomain systemd[1]: Starting Vsftpd ftp daemon...
Sep 28 14:15:40 eavernikovskaya.localdomain systemd[1]: Started Vsftpd ftp daemon.
[root@eavernikovskaya ~]#
```

Рис. 9: Статус службы Very Secure FTP после удаления из автозапуска

Выводим на экран символические ссылки, ответственные за запуск различных сервисов: `ls /etc/systemd/system/multi-user.target.wants`. После ввода этой команды отображается, что ссылки на `vsftpd.service` не существует (рис. 10)

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

Рис. 10: Символические ссылки, ответственные за запуск различных сервисов

Снова добавляем службу Very Secure FTP в автозапуск и опять выводим на экран символические ссылки, ответственные за запуск различных сервисов. На этот раз вывод команды показывает, что создана символическая ссылка для файла `/usr/lib/systemd/system/vsftpd.service` в каталоге `/etc/systemd/system/multi-user.target.wants` (рис. 11), (рис. 12)

```
[root@eavernikovskaya ~]# systemctl enable vsftpd
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /usr/lib/systemd/system/vsftpd.service.
[root@eavernikovskaya ~]#
```

Рис. 11: Добавление службы Very Secure FTP в автозапуск (2)


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

Рис. 12: Символические ссылки, ответственные за запуск различных сервисов после добавления службы в автозапуск

Опять проверяем статус службы Very Secure FTP. Теперь мы видим, что для файла юнита состояние изменено с disabled на enabled (рис. 13)

```
[root@eavernikovskaya ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
   Active: active (running) since Sat 2024-09-28 14:15:40 MSK; 6min ago
     Main PID: 2741 (vsftpd)
        Tasks: 1 (limit: 12205)
      Memory: 708.0K
         CPU: 4ms
       CGroup: /system.slice/vsftpd.service
              └─2741 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 28 14:15:40 eavernikovskaya.localdomain systemd[1]: Starting Vsftpd ftp daemon...
Sep 28 14:15:40 eavernikovskaya.localdomain systemd[1]: Started Vsftpd ftp daemon.
[root@eavernikovskaya ~]#
```

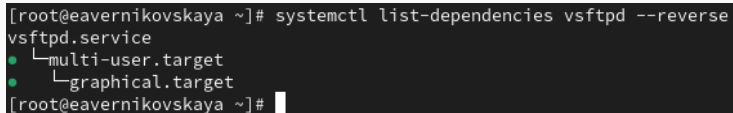
Рис. 13: Статус службы Very Secure FTP (2)

Выводим на экран список зависимостей юнита: *systemctl list-dependencies vsftpd*
(рис. 14)

```
[root@eavernikovskaya ~]# 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
```

Рис. 14: Список зависимостей юнита

Выводим на экран список юнитов, которые зависят от данного юнита:
systemctl list-dependencies vsftpd --reverse (рис. 15)



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

Рис. 15: Список юнитов, которые зависят от данного юнита

Устанавливаем iptables: `dnf -y install iptables*` (рис. 16)

```
[root@eavernikovskaya ~]# dnf -y install iptables*
Last metadata expiration check: 0:10:26 ago on Sat 28 Sep 2024 02:15:00 PM MSK.
Package iptables-libs-1.8.10-2.el9.x86_64 is already installed.
Package iptables-nft-1.8.10-2.el9.x86_64 is already installed.
Dependencies resolved.
=====
Package                        Architecture      Version           Repository        Size
=====
Installing:
iptables-devel                x86_64            1.8.10-4.el9_4    appstream         14 k
iptables-nft-services         noarch            1.8.10-4.el9_4    appstream         17 k
iptables-utils                x86_64            1.8.10-4.el9_4    baseos            40 k
Upgrading:
iptables-libs                 x86_64            1.8.10-4.el9_4    baseos            396 k
iptables-nft                  x86_64            1.8.10-4.el9_4    baseos            186 k
Transaction Summary
=====
Install  3 Packages
Upgrade  2 Packages
```

Рис. 16: Установка iptables

Далее проверяем статус firewalld и iptables (рис. 17), (рис. 18)

```
[root@eavernikovskaya ~]# systemctl status firewalld
• firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
  Active: active (running) since Sat 2024-09-28 14:11:33 MSK; 14min ago
    Docs: man:firewalld(1)
  Main PID: 774 (firewalld)
    Tasks: 2 (limit: 12205)
  Memory: 38.2M
    CPU: 653ms
  CGroup: /system.slice/firewalld.service
          └─774 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid

Sep 28 14:11:32 eavernikovskaya.localdomain systemd[1]: Starting firewalld - dynamic firewall daemon...
Sep 28 14:11:33 eavernikovskaya.localdomain systemd[1]: Started firewalld - dynamic firewall daemon.
[root@eavernikovskaya ~]#
```

Рис. 17: Статус firewalld

```
[root@eavernikovskaya ~]# systemctl status iptables
○ iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; preset: disabled)
   Active: inactive (dead)
[root@eavernikovskaya ~]#
```

Рис. 18: Статус iptables (1)

Далее пробуем запустить firewalld и iptables. При запуске одной службы мы видим, что вторая деактивируется или не запускается (рис. 19)

```
[root@eavernikovskaya ~]# systemctl start firewalld
[root@eavernikovskaya ~]# systemctl start iptables
[root@eavernikovskaya ~]# systemctl status iptables
● iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; preset: disabled)
   Active: active (exited) since Sat 2024-09-28 14:27:07 MSK; 1min 41s ago
     Process: 3180 ExecStart=/usr/libexec/iptables/iptables.init start (code=exited, status=0/SUCCESS)
    Main PID: 3180 (code=exited, status=0/SUCCESS)
      CPU: 20ms

Sep 28 14:27:07 eavernikovskaya.localdomain systemd[1]: Starting IPv4 firewall with iptables...
Sep 28 14:27:07 eavernikovskaya.localdomain iptables.init[3180]: iptables: Applying firewall rules: [ OK ]
Sep 28 14:27:07 eavernikovskaya.localdomain systemd[1]: Finished IPv4 firewall with iptables.
[root@eavernikovskaya ~]# systemctl status firewalld
○ firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
   Active: inactive (dead) since Sat 2024-09-28 14:27:07 MSK; 1min 56s ago
     Duration: 15min 34.116s
       Docs: man:firewalld(1)
     Process: 774 ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS (code=exited, status=0/SUCCESS)
    Main PID: 774 (code=exited, status=0/SUCCESS)
      CPU: 725ms

Sep 28 14:11:32 eavernikovskaya.localdomain systemd[1]: Starting firewalld - dynamic firewall daemon...
Sep 28 14:11:33 eavernikovskaya.localdomain systemd[1]: Started firewalld - dynamic firewall daemon.
Sep 28 14:27:07 eavernikovskaya.localdomain systemd[1]: Stopping firewalld - dynamic firewall daemon...
Sep 28 14:27:07 eavernikovskaya.localdomain systemd[1]: firewalld.service: Deactivated successfully.
Sep 28 14:27:07 eavernikovskaya.localdomain systemd[1]: Stopped firewalld - dynamic firewall daemon.
[root@eavernikovskaya ~]#
```

Рис. 19: Запуск firewalld и iptables и проверка их статусов

Вводим `cat /usr/lib/systemd/system/firewalld.service` (рис. 20)

```
[root@eavernikovskaya ~]# 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 nftables.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

[Install]
WantedBy=multi-user.target
Alias=dbus-org.fedoraproject.FirewallD1.service
[root@eavernikovskaya ~]#
```

Рис. 20: Команда `cat /usr/lib/systemd/system/firewalld.service`

Вводим `cat /usr/lib/systemd/system/iptables.service` (рис. 21)

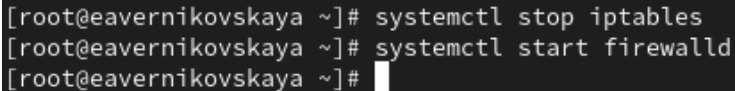
```
[root@eavernikovskaya ~]# cat /usr/lib/systemd/system/iptables.service
[Unit]
Description=IPv4 firewall with iptables
AssertPathExists=/etc/sysconfig/iptables
Before=network-pre.target
Wants=network-pre.target

[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@eavernikovskaya ~]#
```

Рис. 21: Команда `cat /usr/lib/systemd/system/iptables.service`

Выгружаем службу `iptables` (на всякий случай, чтобы убедиться, что данная служба не загружена в систему): `systemctl stop iptables`. После загружаем службу `firewalld` (рис. 22)

A terminal window with a dark background and light-colored text. It shows three lines of commands entered at a root prompt on a machine named 'eavernikovskaya'. The first line is 'systemctl stop iptables', the second is 'systemctl start firewalld', and the third is a blank line with the prompt still visible.

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

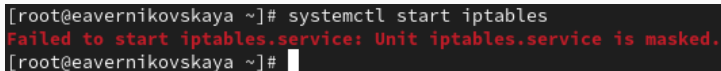
Рис. 22: Выгрузка службы `iptables` и загрузка службы `firewalld`

Далее блокируем запуск `iptables`, введя: `systemctl mask iptables`. При этом будет создана символическая ссылка. Это сделает невозможным случайный запуск сервиса `iptables` (рис. 23)

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

Рис. 23: Блокировка запуска `iptables`

Пробуем запустить iptables. При попытке запустить iptables появляется сообщение об ошибке, указывающее, что служба замаскирована и по этой причине не может быть запущена (рис. 24)



```
[root@eavernikovskaya ~]# systemctl start iptables
Failed to start iptables.service: Unit iptables.service is masked.
[root@eavernikovskaya ~]#
```

Рис. 24: Попытка запустить iptables

Далее пробуем добавить iptables в автозапуск. Сервис будет неактивен, а статус загрузки отобразится как замаскированный (рис. 25) (рис. 26)

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

Рис. 25: Попытка добавить iptables в автозапуск

```
[root@eavernikovskaya ~]# systemctl status iptables
○ iptables.service
   Loaded: masked (Reason: Unit iptables.service is masked.)
   Active: inactive (dead) since Sat 2024-09-28 14:34:12 MSK; 5min ago
     Duration: 7min 4.705s
    Main PID: 3180 (code=exited, status=0/SUCCESS)
       CPU: 86ms

Sep 28 14:27:07 eavernikovskaya.localdomain systemd[1]: Starting IPv4 firewall with iptables...
Sep 28 14:27:07 eavernikovskaya.localdomain iptables.init[3180]: iptables: Applying firewall rules: [ OK ]
Sep 28 14:27:07 eavernikovskaya.localdomain systemd[1]: Finished IPv4 firewall with iptables.
Sep 28 14:34:12 eavernikovskaya.localdomain systemd[1]: Stopping IPv4 firewall with iptables...
Sep 28 14:34:12 eavernikovskaya.localdomain iptables.init[3231]: iptables: Setting chains to policy ACCEPT: raw mangle
Sep 28 14:34:12 eavernikovskaya.localdomain iptables.init[3231]: iptables: Flushing firewall rules: [ OK ]
Sep 28 14:34:12 eavernikovskaya.localdomain systemd[1]: iptables.service: Deactivated successfully.
Sep 28 14:34:12 eavernikovskaya.localdomain systemd[1]: Stopped IPv4 firewall with iptables.
lines 1-15/15 (END)
```

Рис. 26: Статус iptables (2)

Изолируемые цели

Получем список всех активных загруженных целей: `systemctl --type=target`, а затем список всех целей: `systemctl --type=target --all` (рис. 27), (рис. 28)

```
[root@eavernikovskaya ~]# systemctl --type=target
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
basic.target                        loaded active active Basic System
cryptsetup.target                  loaded active active Local Encrypted Volumes
getty.target                        loaded active active Login Prompts
graphical.target                   loaded active active Graphical Interface
integritysetup.target              loaded active active Local Integrity Protected Volumes
local-fs-pre.target                loaded active active Preparation for Local File Systems
local-fs.target                    loaded active active Local File Systems
multi-user.target                  loaded active active Multi-User System
network-online.target              loaded active active Network is Online
network-pre.target                 loaded active active Preparation for Network
network.target                     loaded active active Network
nss-user-lookup.target             loaded active active User and Group Name Lookups
paths.target                       loaded active active Path Units
remote-fs.target                   loaded active active Remote File Systems
slices.target                      loaded active active Slice Units
sockets.target                     loaded active active Socket Units
sound.target                       loaded active active Sound Card
sshd-keygen.target                 loaded active active sshd-keygen.target
swap.target                        loaded active active Swaps
sysinit.target                     loaded active active System Initialization
timers.target                      loaded active active Timer Units
veritysetup.target                 loaded active active Local Verity Protected Volumes

LOAD    = Reflects whether the unit definition was properly loaded.
ACTIVE  = The high-level unit activation state, i.e. generalization of SUB.
SUB     = The low-level unit activation state, values depend on unit type.
22 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
[root@eavernikovskaya ~]#
```

Рис. 27: Список всех активных загруженных целей

Изолируемые цели

```
[root@eavernikovskaya ~]# systemctl --type=target --all
```

UNIT	LOAD	ACTIVE	SUB	DESC
basic.target	loaded	active	active	Basic system targets
blockdev@dev-disk-by\x2duuid-9a39db74\x2d1505\x2d4a34\x2d92d7\x2daba95e8d7183.target	loaded	inactive	dead	Block device target
blockdev@dev-dm\x2d1.target	loaded	inactive	dead	Block device target
blockdev@dev-mapper-rl\x2droot.target	loaded	inactive	dead	Block device target
blockdev@dev-mapper-rl\x2dswap.target	loaded	inactive	dead	Block device target
blockdev@dev-sda1.target	loaded	inactive	dead	Block device target
blockdev@dev-sr0.target	loaded	inactive	dead	Block device target
cryptsetup-pre.target	loaded	inactive	dead	Cryptsetup target
cryptsetup.target	loaded	active	active	Cryptsetup target
display-manager.target	not-found	inactive	dead	Display manager target
emergency.target	loaded	inactive	dead	Emergency target
first-boot-complete.target	loaded	inactive	dead	First boot complete target
getty-pre.target	loaded	inactive	dead	Getty target
getty.target	loaded	active	active	Getty target
graphical.target	loaded	active	active	Graphical target
initrd-fs.target	loaded	inactive	dead	Initrd fs target
initrd-root-device.target	loaded	inactive	dead	Initrd root device target
initrd-root-fs.target	loaded	inactive	dead	Initrd root fs target
initrd-switch-root.target	loaded	inactive	dead	Initrd switch root target
initrd-usr-fs.target	loaded	inactive	dead	Initrd usr fs target
initrd.target	loaded	inactive	dead	Initrd target
integritysetup.target	loaded	active	active	Integrity setup target
local-fs-pre.target	loaded	active	active	Local fs target
local-fs.target	loaded	active	active	Local fs target
multi-user.target	loaded	active	active	Multi-user target
network-online.target	loaded	active	active	Network online target
network-pre.target	loaded	active	active	Network pre target
network.target	loaded	active	active	Network target

Рис. 28: Список всех целей

Переходим в каталог `systemd` и находим список всех целей, которые можно изолировать:

- `cd /usr/lib/systemd/system`
- `grep Isolate *.target*` (рис. 29)

```
[root@eavernikovskaya ~]# cd /usr/lib/systemd/system
[root@eavernikovskaya systemd]# 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@eavernikovskaya systemd]#
```

Далее переключаем операционную систему в режим восстановления: *systemctl isolate rescue.target* (рис. 30), (рис. 31)

```
[root@eavernikovskaya ~]# systemctl isolate reboot.target
```

Рис. 30: Переключение операционной системы в режим восстановления

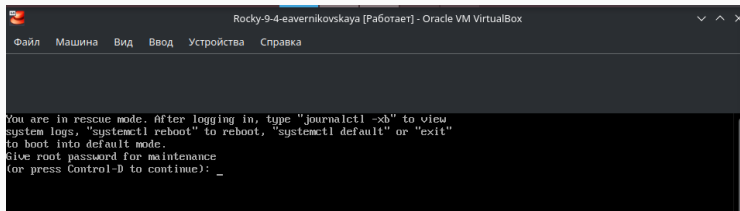
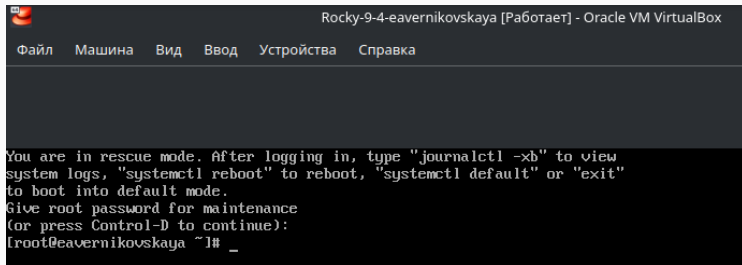


Рис. 31: Режим восстановления

Переходим в режим root и перезапускаем операционную систему: *systemctl isolate reboot.target* (рис. 32), (рис. 33)



```
Rocky-9-4-eavernikovskaya [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка

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.
Give root password for maintenance
(or press Control-D to continue):
[root@eavernikovskaya ~]# _
```

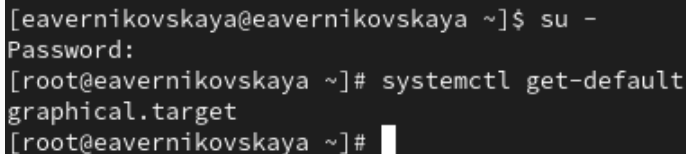
Рис. 32: Режим root

```
[root@eavernikovskaya ~]# systemctl isolate reboot.target
```

Рис. 33: Перезапуск операционной системы

Цель по умолчанию

Получаем права администратора. Далее выводим на экран цель, установленную по умолчанию: *systemctl get-default*. Сейчас графический режим (рис. 34)



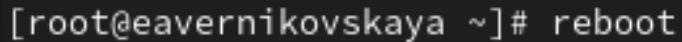
```
[eavernikovskaya@eavernikovskaya ~]$ su -  
Password:  
[root@eavernikovskaya ~]# systemctl get-default  
graphical.target  
[root@eavernikovskaya ~]#
```

Рис. 34: Цель, установленна по умолчанию

Запускаем по умолчанию текстовый режим введя команду *systemctl set-default multi-user.target* и перезагружаем систему командой *reboot* (рис. 35), (рис. 36)

```
[root@eavernikovskaya ~]# 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@eavernikovskaya ~]# █
```

Рис. 35: Запуск по умолчанию текстового режима

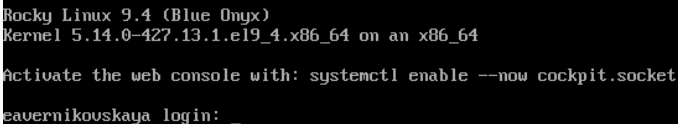
A terminal window with a dark background and light gray text. The prompt is [root@eavernikovskaya ~]# and the command entered is reboot.

```
[root@eavernikovskaya ~]# reboot
```

Рис. 36: Перезагрузка системы (1)

Цель по умолчанию

Система загрузилась в текстовом режиме. Далее получаем полномочия пользователя root и запускаем по умолчанию графический режим введя команду *systemctl set-default graphical.target*. После снова перезагружаем систему командой *reboot* (рис. 37), (рис. 38), (рис. 39), (рис. 40)



```
Rocky Linux 9.4 (Blue Onyx)
Kernel 5.14.0-427.13.1.el9_4.x86_64 on an x86_64

Activate the web console with: systemctl enable --now cockpit.socket

eavernikovskaya login: _
```

Рис. 37: Система в текстовом режиме

```
Rocky Linux 9.4 (Blue Onyx)
Kernel 5.14.0-427.13.1.el9_4.x86_64 on an x86_64

Activate the web console with: systemctl enable --now cockpit.socket

eavernikovskaya login: eavernikovskaya
Password:
Last login: Sat Sep 28 15:00:18 on tty2
[eavernikovskaya@eavernikovskaya ~]$ su -
Password:
[root@eavernikovskaya ~]# _
```

Рис. 38: Учётная запись пользователя root

```
[root@eavernikovskaya ~]# 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@eavernikovskaya ~]#
```

Рис. 39: Запуск по умолчанию графического режима



```
[root@eavernikovskaya ~]# reboot_
```

Рис. 40: Перезагрузка системы (2)

Цель по умолчанию

Система загрузилась в графическом режиме. Всё хорошо! (рис. 41)

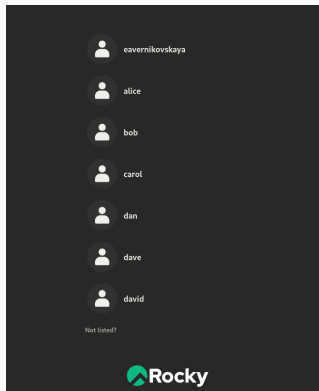


Рис. 41: Система в графическом режиме

Подведение итогов

В ходе выполнения лабораторной работы мы получили навыки управления системными службами операционной системы посредством systemd.

1. Лабораторная работа №5 [Электронный ресурс] URL:
https://esystem.rudn.ru/pluginfile.php/2400698/mod_resource/content/4/006-service.pdf