

# **Лабораторная работа № 5. Управление системными службами**

**Отчёт**

Сергеев Даниил Олегович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Ход выполнения лабораторной работы</b>	<b>7</b>
3.1	Управление сервисами . . . . .	7
3.2	Конфликты юнитов . . . . .	10
3.3	Изолируемые цели . . . . .	13
3.4	Цель по умолчанию . . . . .	14
<b>4</b>	<b>Ответы на контрольные вопросы</b>	<b>16</b>
<b>5</b>	<b>Вывод</b>	<b>18</b>
	<b>Список литературы</b>	<b>19</b>

# Список иллюстраций

3.1	Проверка статуса службы VSFTP до установки . . . . .	7
3.2	Запустим и проверим службу ещё раз . . . . .	8
3.3	Работа с автозапуском службы VSFTP . . . . .	8
3.4	Символические ссылки для различных юнитов . . . . .	9
3.5	Добавление символической ссылки для VSFTP . . . . .	9
3.6	Проверка статуса VSFTP . . . . .	9
3.7	Зависимости VSFTP . . . . .	10
3.8	Установка iptables . . . . .	10
3.9	Проверка статусов firewalld и iptables . . . . .	11
3.10	Запуск двух служб одновременно . . . . .	11
3.11	Файл юнита firewalld . . . . .	12
3.12	Файл юнита iptables . . . . .	12
3.13	Наложение маски на iptables . . . . .	12
3.14	Список активных загруженных целей . . . . .	13
3.15	Список изолируемых целей . . . . .	14
3.16	Окно режима восстановления . . . . .	14
3.17	Установка цели по умолчанию . . . . .	15
3.18	Возврат graphical.target как цели по умолчанию . . . . .	15
3.19	Загрузка системы в графическом режиме . . . . .	15
4.1	Проверка автоматического запуска . . . . .	16
4.2	Вывод всех загруженных сервисных юнитов (первые 9) . . . . .	17

## **Список таблиц**

# 1 Цель работы

Получить навыки управления системными службами операционной системы посредством systemd. [1]

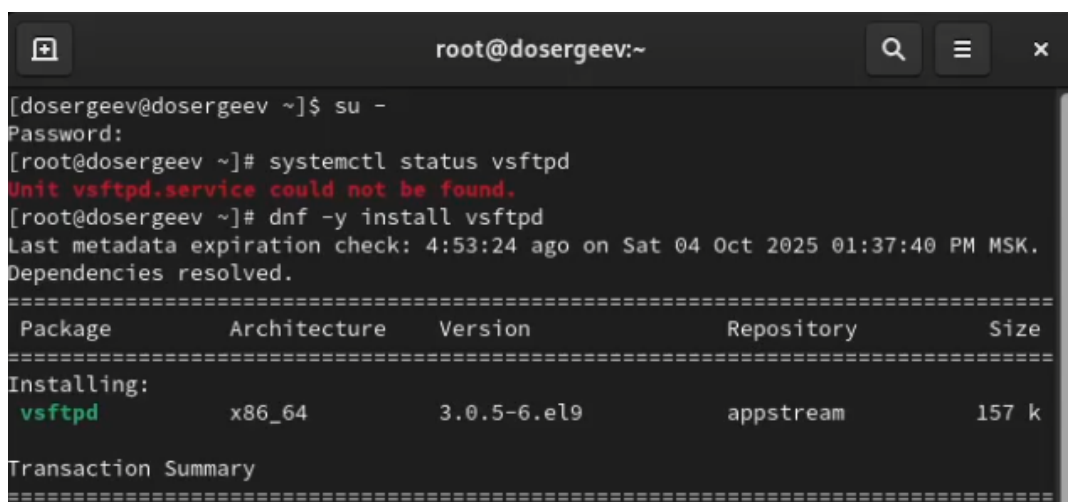
## 2 Задание

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

## 3 Ход выполнения лабораторной работы

### 3.1 Управление сервисами

Откроем терминал под учётной записью root и проверим статус службы VerySecureFTP. Вывод показывает, что юнит vsftpd.service не найден, так как служба ещё не установлена. Поставим VerySecureFTP себе на устройство.



```
root@dosergeev:~  
[dosergeev@dosergeev ~]$ su -  
Password:  
[root@dosergeev ~]# systemctl status vsftpd  
Unit vsftpd.service could not be found.  
[root@dosergeev ~]# dnf -y install vsftpd  
Last metadata expiration check: 4:53:24 ago on Sat 04 Oct 2025 01:37:40 PM MSK.  
Dependencies resolved.  
=====
```

Package	Architecture	Version	Repository	Size
Installing:				
vsftpd	x86_64	3.0.5-6.el9	appstream	157 k

```
Transaction Summary  
=====
```

Рис. 3.1: Проверка статуса службы VSFTP до установки

Запустим службу и проверим её статус. В описании службы указано, что она активна: Active: active (running), но не загружена для автозапуска: Loaded: loaded (...; disabled; ...).

```
root@dosergeev:~  
[root@dosergeev ~]# systemctl start vsftpd  
[root@dosergeev ~]# systemctl status vsftpd  
● vsftpd.service - Vsftpd ftp daemon  
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)  
   Active: active (running) since Sat 2025-10-04 19:15:18 MSK; 6s ago  
     Process: 3290 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exited, status=0/SUCCESS)  
    Main PID: 3291 (vsftpd)  
       Tasks: 1 (limit: 48749)  
      Memory: 740.0K  
         CPU: 7ms  
      CGroup: /system.slice/vsftpd.service  
              └─3291 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf  
  
Oct 04 19:15:18 dosergeev.localdomain systemd[1]: Starting Vsftpd ftp daemon...  
Oct 04 19:15:18 dosergeev.localdomain systemd[1]: Started Vsftpd ftp daemon.  
[root@dosergeev ~]#
```

Рис. 3.2: Запустим и проверим службу ещё раз

Добавим службу VSFTP в автозапуск, для этого используем `systemctl enable vsftpd`, затем проверим статус. В строке `Loaded` стутс `disabled` сменился на `enabled`. Теперь удалим службу из автозапуска, написав `systemctl disable vsftpd`. Ещё раз проверим статус - теперь в строке `Loaded` снова написано `disabled`.

```
[root@dosergeev ~]# systemctl enable vsftpd  
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /usr/lib/systemd/system/vsftpd.s  
ervice.  
[root@dosergeev ~]# systemctl status vsftpd  
● vsftpd.service - Vsftpd ftp daemon  
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)  
   Active: active (running) since Sat 2025-10-04 19:15:18 MSK; 36s ago  
    Main PID: 3291 (vsftpd)  
       Tasks: 1 (limit: 48749)  
      Memory: 740.0K  
         CPU: 7ms  
      CGroup: /system.slice/vsftpd.service  
              └─3291 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf  
  
Oct 04 19:15:18 dosergeev.localdomain systemd[1]: Starting Vsftpd ftp daemon...  
Oct 04 19:15:18 dosergeev.localdomain systemd[1]: Started Vsftpd ftp daemon.  
[root@dosergeev ~]# systemctl disable vsftpd  
Removed "/etc/systemd/system/multi-user.target.wants/vsftpd.service".  
[root@dosergeev ~]# systemctl status vsftpd  
● vsftpd.service - Vsftpd ftp daemon  
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)  
   Active: active (running) since Sat 2025-10-04 19:15:18 MSK; 47s ago  
    Main PID: 3291 (vsftpd)  
       Tasks: 1 (limit: 48749)  
      Memory: 740.0K  
         CPU: 7ms  
      CGroup: /system.slice/vsftpd.service  
              └─3291 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf  
  
Oct 04 19:15:18 dosergeev.localdomain systemd[1]: Starting Vsftpd ftp daemon...  
Oct 04 19:15:18 dosergeev.localdomain systemd[1]: Started Vsftpd ftp daemon.  
[root@dosergeev ~]#
```

Рис. 3.3: Работа с автозапуском службы VSFTP

Выведем на экран символические ссылки, ответственные за запуск различных сервисов, для этого напишем команду `ls /etc/systemd/system/multi-`



user.target.wants. Заметим, что ссылки на vsftpd.service нет. Дополнительно проверим это, используя grep -i vsftp для вывода команды.

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

Рис. 3.4: Символические ссылки для различных юнитов

Снова добавим VSFTP в автозапуск и выведем символические ссылки, используя grep. Теперь в каталоге multi-user.target.wants находится символическая ссылка на vsftpd.service.

```
[root@dosergeev ~]# systemctl enable vsftpd
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /usr/lib/systemd/system/vsftpd.service.
[root@dosergeev ~]# ls -l /etc/systemd/system/multi-user.target.wants/ | grep -i vsftpd
lrwxrwxrwx. 1 root root 38 Oct  4 19:17 vsftpd.service → /usr/lib/systemd/system/vsftpd.service
[root@dosergeev ~]#
```

Рис. 3.5: Добавление символической ссылки для VSFTP

Снова проверим статус VSFTP и проверим ещё раз строку с автозапуском. Теперь в Loaded указан статус enabled.

```
[root@dosergeev ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
   Active: active (running) since Sat 2025-10-04 19:15:18 MSK; 3min 0s ago
     Main PID: 3291 (vsftpd)
       Tasks: 1 (limit: 48749)
      Memory: 740.0K
         CPU: 7ms
       CGroup: /system.slice/vsftpd.service
               └─3291 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Oct 04 19:15:18 dosergeev.localdomain systemd[1]: Starting Vsftpd ftp daemon...
Oct 04 19:15:18 dosergeev.localdomain systemd[1]: Started Vsftpd ftp daemon.
[root@dosergeev ~]#
```

Рис. 3.6: Проверка статуса VSFTP

Выведем на экран зависимости юнита VSFTP и, наоборот, список юнитов, которые зависят от него.

```
[root@dosergeev ~]# systemctl list-dependencies vsftpd | head -10
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
└─
```

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

Рис. 3.7: Зависимости VSFTP

## 3.2 Конфликты юнитов

Установим пакеты iptables.

```
root@dosergeev:~
[root@dosergeev ~]# dnf -y install iptables\*
Last metadata expiration check: 0:05:13 ago on Sat 04 Oct 2025 06:31:12 PM MSK.
Package iptables-libs-1.8.10-11.el9_5.x86_64 is already installed.
Package iptables-nft-1.8.10-11.el9_5.x86_64 is already installed.
Dependencies resolved.
=====
Package                                Architecture    Version          Repository      Size
=====
Installing:
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
=====
Transaction Summary
=====
```

Рис. 3.8: Установка iptables

Проверим статусы firewalld и iptables. Один из них запущен (firewalld), а другой неактивен (iptables).

```

[root@dosergeev ~]# 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 Sat 2025-10-04 18:56:44 MSK; 29min ago
     Docs: man:firewalld(1)
    Main PID: 868 (firewalld)
      Tasks: 2 (limit: 48749)
     Memory: 43.6M
        CPU: 1.061s
    CGroup: /system.slice/firewalld.service
            └─868 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid

Oct 04 18:56:42 dosergeev.localdomain systemd[1]: Starting firewalld - dynamic firewall daemon...
Oct 04 18:56:44 dosergeev.localdomain systemd[1]: Started firewalld - dynamic firewall daemon.
[root@dosergeev ~]# systemctl status iptables
● iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; preset: disabled)
   Active: inactive (dead)
[root@dosergeev ~]#

```

Рис. 3.9: Проверка статусов firewalld и iptables

Попробуем запустить эти службы по очереди, проверяя их статус. После запуска iptables, юнит firewalld стал неактивен, так как он несовместим.

```

[root@dosergeev ~]# systemctl start firewalld
[root@dosergeev ~]# systemctl start iptables
[root@dosergeev ~]# 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 2025-10-04 19:26:38 MSK; 5s ago
     Duration: 1.004s
     Docs: man:firewalld(1)
    Process: 3712 ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS (code=exited, status=0/SUCCESS)
    Main PID: 3712 (code=exited, status=0/SUCCESS)
        CPU: 552ms

Oct 04 19:26:37 dosergeev.localdomain systemd[1]: Starting firewalld - dynamic firewall daemon...
Oct 04 19:26:37 dosergeev.localdomain systemd[1]: Started firewalld - dynamic firewall daemon.
Oct 04 19:26:38 dosergeev.localdomain systemd[1]: Stopping firewalld - dynamic firewall daemon...
Oct 04 19:26:38 dosergeev.localdomain systemd[1]: firewalld.service: Deactivated successfully.
Oct 04 19:26:38 dosergeev.localdomain systemd[1]: Stopped firewalld - dynamic firewall daemon.
[root@dosergeev ~]# 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 2025-10-04 19:26:38 MSK; 11s ago
     Process: 3783 ExecStart=/usr/libexec/iptables/iptables.init start (code=exited, status=0/SUCCESS)
    Main PID: 3783 (code=exited, status=0/SUCCESS)
        CPU: 33ms

Oct 04 19:26:38 dosergeev.localdomain systemd[1]: Starting IPv4 firewall with iptables...
Oct 04 19:26:38 dosergeev.localdomain iptables.init[3783]: iptables: Applying firewall rules: [ OK ]
Oct 04 19:26:38 dosergeev.localdomain systemd[1]: Finished IPv4 firewall with iptables.
[root@dosergeev ~]#

```

Рис. 3.10: Запуск двух служб одновременно

Проверим зависимости этих служб, открыв юниты командами `cat /usr/lib/systemd/system/firewalld.service` и `cat /usr/lib/systemd/system/iptables.service`. Служба iptables не имеет конфликтных сервисов, однако в firewalld указаны конфликты: iptables.service, ip6tables.service, ebtables.service, ipset.service.

```
[root@dosergeev ~]# 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
# 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@dosergeev ~]#
```

Рис. 3.11: Файл юнита firewalld

```
[root@dosergeev ~]# 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@dosergeev ~]#
```

Рис. 3.12: Файл юнита iptables

Выгрузим службу iptables и загрузим firewalld. Заблокируем запуск iptables, наложив на него маску. Попробуем снова запустить и добавить в автозапуск службу iptables. Обе попытки выдадут ошибку, которая указывает что сервис находится под маской.

```
[root@dosergeev ~]# systemctl mask iptables
Created symlink /etc/systemd/system/iptables.service → /dev/null.
[root@dosergeev ~]# ls -l /etc/systemd/system/ | grep -i iptables
lrwxrwxrwx. 1 root root 9 Oct 4 19:28 iptables.service -> /dev/null
[root@dosergeev ~]# systemctl start iptables
Failed to start iptables.service: Unit iptables.service is masked.
[root@dosergeev ~]# systemctl enable iptables
Failed to enable unit: Unit file /etc/systemd/system/iptables.service is masked.
[root@dosergeev ~]#
```

Рис. 3.13: Наложение маски на iptables

### 3.3 Изолируемые цели

Получим список всех активных загруженных целей, введя `systemctl -type=target`. Теперь выведем список всех целей с помощью `systemctl -type=target -all`.

```
[root@dosergeev ~]# 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@dosergeev ~]#
```

Рис. 3.14: Список активных загруженных целей

Перейдем в каталог `systemd` и проверим, какие цели можно изолировать, используя команду `grep Isolate *.target`.

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

Рис. 3.15: Список изолируемых целей

Переключим операционную систему в режим восстановления: `systemctl isolate rescue.target`. Выполнение данной команды прекратит дургие службы кроме `rescue` и её зависимостей.

```
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@dosergeev ~]# _
```

Рис. 3.16: Окно режима восстановления

Введем пароль `root`, чтобы войти в систему, и перезапустим ОС.

### 3.4 Цель по умолчанию

Войдем в терминал под учетной записью администратора. Выведем на экран цель по умолчанию: `system get-default`. Получим в выводе службу `graphical.target`. Установим для запуска по умолчанию текстовый режим, указав в опции команды `systemctl set-default` юнит `multi-user.target`.

```
[dosergeev@dosergeev ~]$ systemctl get-default
graphical.target
[dosergeev@dosergeev ~]$ 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.
[dosergeev@dosergeev ~]$ systemctl get-default
multi-user.target
[dosergeev@dosergeev ~]$
```

Рис. 3.17: Установка цели по умолчанию

Перезагрузим ОС. Система запустилась в текстовом режиме. Войдем в учетную запись root и вернем цель по умолчанию graphical.target. Снова перезапустим ОС.

```
Rocky Linux 9.6 (Blue Onyx)
Kernel 5.14.0-570.39.1.el9_6.x86_64 on x86_64

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

dosergeev login: root
Password:
Last login: Sat Oct 4 19:12:15 on pts/0
[root@dosergeev ~]# 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@dosergeev ~]# systemctl get-default
graphical.target
[root@dosergeev ~]# reboot_
```

Рис. 3.18: Возврат graphical.target как цели по умолчанию

Система загрузилась в графическом режиме.

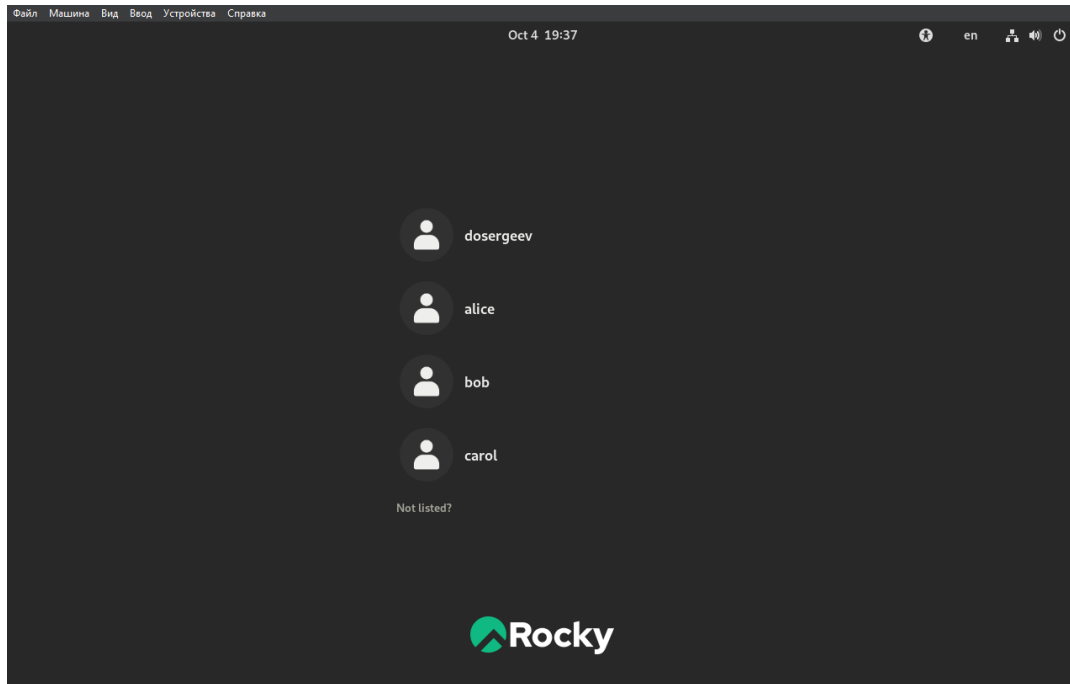


Рис. 3.19: Загрузка системы в графическом режиме

## 4 Ответы на контрольные вопросы

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

- Юнит(unit) - это объекты конфигурации, используются для выполнения операций над системными службами systemd. Они бывают разных типов. Например, существует `service` (управляет запуском системных служб), `timer` (аналог `cron` для планирования задач), `target` (объединение нескольких сервисов) и прочие.
- Пример: `graphical.target` используется для запуска графического интерфейса, `sshd.service` задает параметры для демона (процесса) `sshd`, тем самым управляя им.

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

- `systemctl is-enabled <цель>` - узнать напрямую
- `systemctl status <цель> | grep -i loaded` - получить через статус цели

```
[dosergeev@dosergeev ~]$ systemctl is-enabled firewalld
enabled
[dosergeev@dosergeev ~]$ systemctl status firewalld | grep -i loaded
Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
[dosergeev@dosergeev ~]$
```

Рис. 4.1: Проверка автоматического запуска

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



- `systemctl list-units --type=service` - все юниты типа service

```
[dosergeev@dosergeev ~]$ systemctl list-units --type=service | head -n 10
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
accounts-daemon.service            loaded active running Accounts Service
alsa-state.service                 loaded active running Manage Sound Card State (restore and store)
atd.service                         loaded active running Deferred execution scheduler
auditd.service                     loaded active running Security Auditing Service
avahi-daemon.service               loaded active running Avahi mDNS/DNS-SD Stack
chronyd.service                     loaded active running NTP client/server
colord.service                      loaded active running Manage, Install and Generate Color Profiles
crond.service                       loaded active running Command Scheduler
cups.service                        loaded active running CUPS Scheduler
[dosergeev@dosergeev ~]$
```

Рис. 4.2: Вывод всех загруженных сервисных юнитов (первые 9)

#### 4. Как создать потребность (wants) в сервисе?

- `systemctl add-wants <цель>.target <сервис>.service` - автоматическое добавление через systemctl
- `ln -s /usr/lib/systemd/system/<зависимый сервис>.service /etc/systemd/system/<сервис>.service.wants/` - добавление символической ссылки вручную

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

- `systemctl rescue`
- `systemctl isolate rescue.target`

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

- Цель не может быть изолирована, если в ней отсутствует параметр `AllowIsolate=yes`, либо его значение равно `AllowIsolate=no`

#### 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?

- `systemctl list-dependencies <цель>`

## **5 Вывод**

В результате выполнения лабораторной работы я ознакомился с инструментами управления системными службами операционной системы, выполнил основные операции со службами на примере VerySecureFTP, научился разрешать конфликты юнитов и изолировать их.

## Список литературы

1. Kulyabov, Korolykova. Лабораторная работа № 5. Управление системными службами. [https://esystem.rudn.ru/pluginfile.php/2843472/mod\\_resource/content/4/006-service.pdf](https://esystem.rudn.ru/pluginfile.php/2843472/mod_resource/content/4/006-service.pdf); RUDN.