

Лабораторная работа №5

Управление системными службами

Комягин Андрей Николаевич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Управление сервисами	6
2.2	Конфликты юнитов	9
2.3	Изолируемые цели	12
2.4	Цель по умолчанию	13
3	Контрольные вопросы	14
3.1	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?	15
4	Вывод	16
	Список литературы	17

Список иллюстраций

2.1	Установка службы Very Secure FTP	6
2.2	Статус Very Secure FTP	6
2.3	автозапуск	7
2.4	символические ссылки	7
2.5	Список зависимостей	8
2.6	Список юнитов, зависящих от юнита	9
2.7	установка iptables	9
2.8	запуск конфликтующих юнитов	10
2.9	Конфликты юнитов	11
2.10	блокировка iptables	12
2.11	Изолируемые цели	12
2.12	режим восстановления	12
2.13	цель по умолчанию	13
2.14	режимы системы	13

Список таблиц

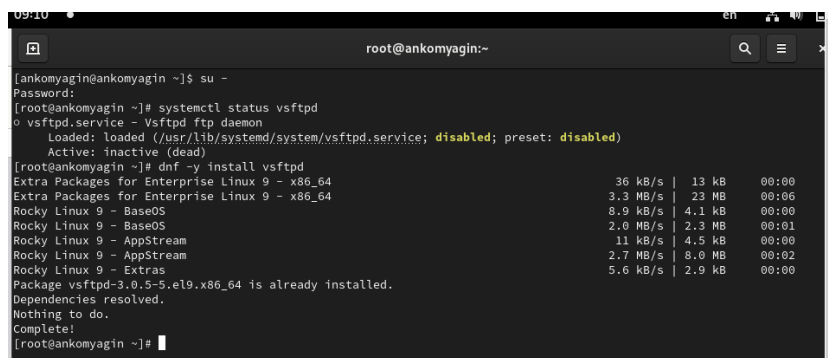
1 Цель работы

Получить навыки управления системными службами операционной системы посредством systemd.

2 Выполнение лабораторной работы

2.1 Управление сервисами

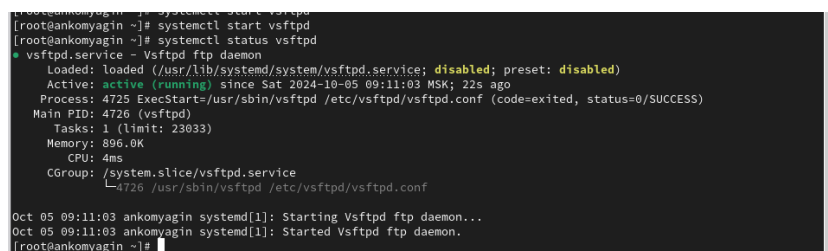
Получим полномочия администратора, проверим статус службы Very Secure FTP, Установим службу Very Secure FTP (рис. 2.1).



```
09:10 root@ankomyagin:~  
[ankomyagin@ankomyagin ~]$ su -  
Password:  
[root@ankomyagin ~]# systemctl status vsftpd  
● vsftpd.service - Vsftpd ftp daemon  
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)  
   Active: inactive (dead)  
[root@ankomyagin ~]# dnf -y install vsftpd  
Extra Packages for Enterprise Linux 9 - x86_64      36 kB/s | 13 kB  00:00  
Extra Packages for Enterprise Linux 9 - x86_64      3.3 MB/s | 23 MB 00:06  
Rocky Linux 9 - BaseOS                             8.9 kB/s | 4.1 kB 00:00  
Rocky Linux 9 - BaseOS                             2.0 MB/s | 2.3 MB 00:01  
Rocky Linux 9 - AppStream                          11 kB/s | 4.5 kB 00:00  
Rocky Linux 9 - AppStream                          2.7 MB/s | 8.0 MB 00:02  
Rocky Linux 9 - Extras                             5.6 kB/s | 2.9 kB 00:00  
Package vsftpd-3.0.5-5.el9.x86_64 is already installed.  
Dependencies resolved.  
Nothing to do.  
Complete!  
[root@ankomyagin ~]#
```

Рис. 2.1: Установка службы Very Secure FTP

Запустим службу Very Secure FTP. Проверим статус службы Very Secure FTP (рис. 2.2).



```
[root@ankomyagin ~]# systemctl start vsftpd  
[root@ankomyagin ~]# systemctl status vsftpd  
● vsftpd.service - Vsftpd ftp daemon  
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)  
   Active: active (running) since Sat 2024-10-05 09:11:03 MSK; 22s ago  
     Process: 4725 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exited, status=0/SUCCESS)  
    Main PID: 4726 (vsftpd)  
       Tasks: 1 (limit: 23033)  
      Memory: 896.0K  
         CPU: 4ms  
    CGroup: /system.slice/vsftpd.service  
            └─4726 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf  
  
Oct 05 09:11:03 ankomyagin systemd[1]: Starting Vsftpd ftp daemon...  
Oct 05 09:11:03 ankomyagin systemd[1]: Started Vsftpd ftp daemon.  
[root@ankomyagin ~]#
```

Рис. 2.2: Статус Very Secure FTP

Добавим службу Very Secure FTP в автозапуск при загрузке операционной

системы. Удалим службу из автозапуска (рис. 2.3).

```
root@ankomyagin:~  
[root@ankomyagin ~]# systemctl enable vsftpd  
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /usr/lib/systemd/system/vsftpd.service.  
[root@ankomyagin ~]# systemctl status vsftpd  
● vsftpd.service - Vsftpd ftp daemon  
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)  
   Active: active (running) since Sat 2024-10-05 09:11:03 MSK; 1min 25s ago  
     Main PID: 4726 (vsftpd)  
       Tasks: 1 (limit: 23033)  
      Memory: 896.0K  
         CPU: 4ms  
    CGroup: /system.slice/vsftpd.service  
            └─4726 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf  
  
Oct 05 09:11:03 ankomyagin systemd[1]: Starting Vsftpd ftp daemon...  
Oct 05 09:11:03 ankomyagin systemd[1]: Started Vsftpd ftp daemon.  
[root@ankomyagin ~]# systemctl disable  
Too few arguments.  
[root@ankomyagin ~]# systemctl disable vsftpd  
Removed "/etc/systemd/system/multi-user.target.wants/vsftpd.service".  
[root@ankomyagin ~]# systemctl status vsftpd  
● vsftpd.service - Vsftpd ftp daemon  
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)  
   Active: active (running) since Sat 2024-10-05 09:11:03 MSK; 1min 51s ago  
     Main PID: 4726 (vsftpd)  
       Tasks: 1 (limit: 23033)  
      Memory: 896.0K  
         CPU: 4ms  
    CGroup: /system.slice/vsftpd.service  
            └─4726 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf  
  
Oct 05 09:11:03 ankomyagin systemd[1]: Starting Vsftpd ftp daemon...  
Oct 05 09:11:03 ankomyagin systemd[1]: Started Vsftpd ftp daemon.  
[root@ankomyagin ~]#
```

Рис. 2.3: автозапуск

Выведем на экран символические ссылки, ответственные за запуск различных сервисов, добавим службу Very Secure FTP в автозапуск и выведем на экран символические ссылки(рис. 2.4).

```
root@ankomyagin:~  
[root@ankomyagin ~]# 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 firewalld.service    mdmmonitor.service     smartd.service        vboxadd-service.service  
chronyd.service      irqbalance.service  ModemManager.service  sshd.service          vmttoolsd.service  
crond.service        kdump.service       NetworkManager.service sssd.service          vsftpd.service  
  
[root@ankomyagin ~]# systemctl enable vsftpd  
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /usr/lib/systemd/system/vsftpd.service.  
[root@ankomyagin ~]# 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 firewalld.service    mdmmonitor.service     smartd.service        vboxadd-service.service  
chronyd.service      irqbalance.service  ModemManager.service  sshd.service          vmttoolsd.service  
crond.service        kdump.service       NetworkManager.service sssd.service          vsftpd.service  
  
[root@ankomyagin ~]# systemctl status vsftpd  
● vsftpd.service - Vsftpd ftp daemon  
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)  
   Active: active (running) since Sat 2024-10-05 09:11:03 MSK; 3min 49s ago  
     Main PID: 4726 (vsftpd)  
       Tasks: 1 (limit: 23033)  
      Memory: 896.0K  
         CPU: 4ms  
    CGroup: /system.slice/vsftpd.service  
            └─4726 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf  
  
Oct 05 09:11:03 ankomyagin systemd[1]: Starting Vsftpd ftp daemon...  
Oct 05 09:11:03 ankomyagin systemd[1]: Started Vsftpd ftp daemon.  
[root@ankomyagin ~]#
```

Рис. 2.4: символические ссылки

Выведем на экран список зависимостей юнита (рис. 2.5).

```

[root@ankomyagin ~]# 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-pcrmachine.service
○   ├──systemd-pcrphase-sysinit.service
○   ├──systemd-pcrphase.service
●   ├──systemd-random-seed.service
○   ├──systemd-repart.service
●   ├──systemd-sysctl.service
○   ├──systemd-sysusers.service
●   ├──systemd-tmpfiles-setup-dev.service
●   ├──systemd-tmpfiles-setup.service
●   ├──systemd-udev-trigger.service
●   ├──systemd-udevd.service
○   ├──systemd-update-done.service
●   ├──systemd-update-utmp.service
●   ├──cryptsetup.target
●   ├──integritysetup.target
●   ├──local-fs.target

```

Рис. 2.5: Список зависимостей

Выведем на экран список юнитов, которые зависят от данного юнита(рис. 2.6)


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

Рис. 2.6: Список юнитов, зависящих от юнита

2.2 Конфликты юнитов

Некоторые юниты могут конфликтовать друг с другом и, соответственно, не могут работать одновременно

Установим iptables (рис. 2.7)

```
[root@anonymagrin ~]# dnf -y install iptables-libs --skip-broken
Last metadata expiration check: 0:09:52 ago on Sat 05 Oct 2024 09:08:10 AM MSK.
Package iptables-libs-1.8.10-4.el9_4.x86_64 is already installed.
Package iptables-nft-1.8.10-4.el9_4.x86_64 is already installed.
Dependencies resolved.

Problem: conflicting requests
- nothing provides iptables-libs(x86-64) = 1.8.10-2.el9 or iptables-libs(x86-64) = 1.8.10-2.el9_1 needed by iptables-legacy-1.8.10-2.2.el9.x86_64 from epel
=====
Package                               Architecture      Version           Repository        Size
=====
Installing:
iptables-devel                       x86_64            1.8.10-4.el9_4    appstream         14 kB
iptables-legacy-devel               x86_64            1.8.10-2.2.el9    epel              13 kB
iptables-legacy-libs                x86_64            1.8.10-2.2.el9    epel              37 kB
iptables-nft-services                noarch            1.8.10-4.el9_4    appstream         17 kB
iptables-services                    noarch            1.8.10-2.2.el9    epel              15 kB
iptables-utils                       x86_64            1.8.10-4.el9_4    baseos            40 kB
Skipping packages with broken dependencies:
iptables-legacy                     x86_64            1.8.10-2.2.el9    epel              49 kB

Transaction Summary
-----
Install 6 Packages
Skip    1 Package

Total download size: 137 k
Installed size: 272 k
Downloading Packages:
(1/6): iptables-legacy-devel-1.8.10-2.2.el9.x86_64.rpm          58 kB/s | 13 KB   00:00
(2/6): iptables-services-1.8.10-2.2.el9.noarch.rpm             57 kB/s | 15 KB   00:00
(3/6): iptables-legacy-libs-1.8.10-2.2.el9.x86_64.rpm         110 kB/s | 37 KB   00:00
(4/6): iptables-utils-1.8.10-4.el9_4.x86_64.rpm               209 kB/s | 40 KB   00:00
(5/6): iptables-nft-services-1.8.10-4.el9_4.noarch.rpm        120 kB/s | 17 KB   00:00
(6/6): iptables-devel-1.8.10-4.el9_4.x86_64.rpm               91 kB/s | 14 KB   00:00
-----
Total                                                                68 kB/s | 137 KB   00:02
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
Installing : iptables-legacy-libs-1.8.10-2.2.el9.x86_64       1/1
Installing : iptables-legacy-devel-1.8.10-2.2.el9.x86_64     1/6
Installing : iptables-devel-1.8.10-4.el9_4.x86_64            2/6
Installing : iptables-nft-services-1.8.10-4.el9_4.noarch      3/6
```

Рис. 2.7: установка iptables

Проверим статус `firewalld` и `iptables`, попробуем запустить `firewalld` и `iptables` (рис. 2.8).

```
09:20 • root@ankomyagin:~  
[root@ankomyagin ~]# 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 2024-10-05 09:05:02 MSK; 14min ago  
     Docs: man:firewalld(1)  
    Main PID: 886 (firewalld)  
      Tasks: 2 (limit: 23033)  
    Memory: 39.4M  
       CPU: 358ms  
    CGroup: /system.slice/firewalld.service  
            └─886 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid  
  
Oct 05 09:05:02 ankomyagin systemd[1]: Starting firewalld - dynamic firewall daemon...  
Oct 05 09:05:02 ankomyagin systemd[1]: Started firewalld - dynamic firewall daemon.  
[root@ankomyagin ~]# systemctl status iptables  
○ iptables.service - IPv4 firewall with iptables  
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; preset: disabled)  
   Active: inactive (dead)  
[root@ankomyagin ~]# systemctl start firewalld  
[root@ankomyagin ~]# systemctl start iptables  
[root@ankomyagin ~]# systemctl status firewalld~  
Invalid unit name "firewalld~" escaped as "firewalld\x7e" (maybe you should use systemd-escape?).  
Unit firewalld\x7e.service could not be found.  
[root@ankomyagin ~]# 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 2024-10-05 09:19:52 MSK; 26s ago  
 Duration: 14min 49.259s  
     Docs: man:firewalld(1)  
 Process: 886 ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS (code=exited, status=0/SUCCESS)  
    Main PID: 886 (code=exited, status=0/SUCCESS)  
       CPU: 411ms  
  
Oct 05 09:05:02 ankomyagin systemd[1]: Starting firewalld - dynamic firewall daemon...  
Oct 05 09:05:02 ankomyagin systemd[1]: Started firewalld - dynamic firewall daemon...  
Oct 05 09:19:52 ankomyagin systemd[1]: Stopping firewalld - dynamic firewall daemon...  
Oct 05 09:19:52 ankomyagin systemd[1]: firewalld.service: Deactivated successfully.  
Oct 05 09:19:52 ankomyagin systemd[1]: Stopped firewalld - dynamic firewall daemon.  
[root@ankomyagin ~]#
```

Рис. 2.8: запуск конфликтующих юнитов

Конфликты юнитов(рис. 2.9)

```
root@ankomyagin:~  
[root@ankomyagin ~]# 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  
# 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@ankomyagin ~]# 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@ankomyagin ~]#
```

Рис. 2.9: Конфликты юнитов

1. Настройки конфликтов **firewalld.service**

- `Conflicts=iptables.service ip6tables.service ebtables.service ipset.service nftables.service`
- Это означает, что `firewalld` не может работать одновременно с `iptables` и другими упомянутыми сервисами. Если `firewalld` запущен, то все перечисленные сервисы будут остановлены.

2. Настройки конфликтов **/iptables.service**

- Отсутствие явных конфликтов в этом файле, но подразумевается, что он будет конфликтовать с `firewalld`, поскольку оба сервиса пытаются управлять правилами межсетевого экрана.

Выгрузим службу `iptables`, заблокируем запуск `iptables`, попробуем запустить `iptables`. Попробуем добавить `iptables` в автозапуск(рис. 2.10)

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

Рис. 2.10: блокировка iptables

2.3 Изолируемые цели

Перейдем в каталог systemd и найдите список всех целей, которые можно изолировать(рис. 2.11)

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

Рис. 2.11: Изолируемые цели

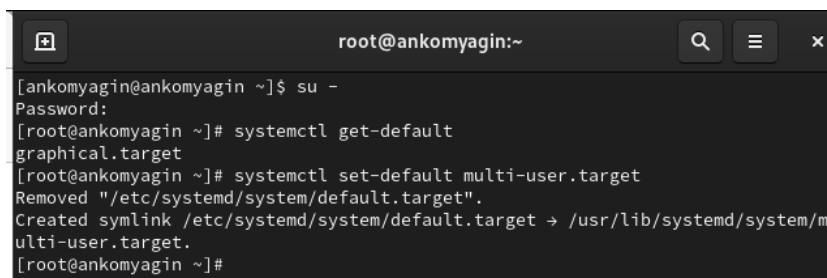
Переключим операционную систему в режим восстановления(рис. 2.12)

```
system-update.target:AllowIsolate=yes  
[root@ankomyagin system]# systemctl isolate rescue.target
```

Рис. 2.12: режим восстановления

2.4 Цель по умолчанию

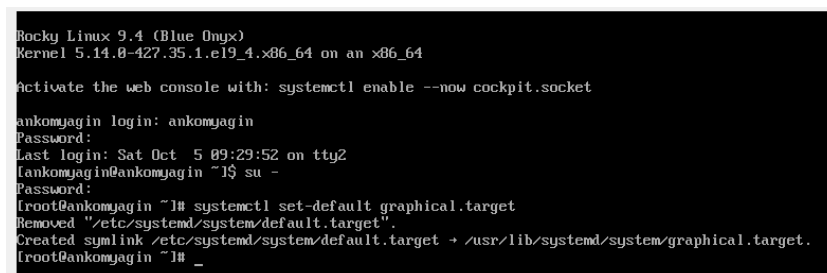
Выведем на экран цель, установленную по умолчанию. (рис. 2.13)



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

Рис. 2.13: цель по умолчанию

Перегрузим систему командой `reboot`. Вновь перегрузим систему командой `reboot`. Убедимся, что система загрузилась в графическом режиме(рис. 2.14)



```
Rocky Linux 9.4 (Blue Onyx)  
Kernel 5.14.0-427.35.1.el9_4.x86_64 on an x86_64  
  
Activate the web console with: systemctl enable --now cockpit.socket  
  
ankomyagin login: ankomyagin  
Password:  
Last login: Sat Oct 5 09:29:52 on tty2  
[ankomyagin@ankomyagin ~]$ su -  
Password:  
[root@ankomyagin ~]# 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@ankomyagin ~]#
```

Рис. 2.14: режимы системы

3 Контрольные вопросы

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

Юнит в контексте systemd — это абстракция, представляющая собой объект, который управляется системой и может быть запущен или остановлен. Существует несколько типов юнитов, включая:

- Сервисные юниты (service): представляют собой службы, которые выполняют определенные задачи (например, `httpd.service` для Apache).
- Целевые юниты (target): группы других юнитов, которые могут быть активированы вместе (например, `multi-user.target`).
- Монтажные юниты (mount): представляют собой точки монтирования файловых систем.
- Сокетные юниты (socket): управляют сокетами для межпроцессного взаимодействия.

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

Для этого используется команда:

`sudo systemctl disable`

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

Для отображения всех загруженных сервисных юнитов используйте команду:

systemctl list-units --type=service

4. Как создать потребность (wants) в сервисе?

Для создания зависимости типа “wants” можно использовать команду:

sudo systemctl add-wants

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

Для переключения на цель восстановления используйте команду:

sudo systemctl isolate rescue.target

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

Сообщение о том, что цель не может быть изолирована, может возникнуть, если цель имеет активные зависимости или другие юниты, которые не могут быть остановлены без нарушения работы системы. Это может происходить из-за активных служб или процессов, которые требуют других юнитов.

3.1 7. Вы хотите отключить службу systemd, но, прежде чем сделать это, вы хотите узнать, какие другие юниты зависят от этой службы. Какую команду вы бы использовали?

Для отображения зависимостей используйте команду:

****systemctl list-dependencies <имя_службы>****

4 Вывод

В ходе выполнения лабораторной работы я получил навыки управления системными службами операционной системы посредством `systemd`.

Список литературы

Туис, курс Администрирование операционных систем