

Отчёт по лабораторной работе №10

Основы работы с модулями ядра операционной системы

Акунаева Антонина Эрдниевна

Содержание

1 Цель работы	5
2 Задание	6
3 Выполнение лабораторной работы	7
4 Контрольные вопросы	15
5 Выводы	19
Список литературы	20

Список иллюстраций

3.1 Отображение информации об устройствах системы и модулях	8
3.2 Отображение списка загруженных модулей ядра	8
3.3 Проверка загрузки модуля ядра (ext4)	9
3.4 Выгрузка модулей ядра (ext4 и xfs)	10
3.5 Загрузка модуля ядра bluetooth	11
3.6 Параметры bluetooth и выгрузка модуля	12
3.7 Информация о текущей версии ядра и доступных пакетах	12
3.8 Обновление системы	13
3.9 Обновление ядра и ОС	13
3.10 Проверка установки новой версии ядра ОС	14
4.1 Контрольный вопрос №2	15
4.2 Контрольный вопрос №3	16
4.3 Контрольный вопрос №4-1	16
4.4 Контрольный вопрос №4-2	17
4.5 Контрольный вопрос №5	17
4.6 Контрольный вопрос №6	17
4.7 Контрольный вопрос №7	18
4.8 Контрольный вопрос №8	18

Список таблиц

1 Цель работы

Получить навыки работы с утилитами управления модулями ядра операционной системы. [1]

2 Задание

1. Продемонстрируйте навыки работы по управлению модулями ядра (см. раздел 10.4.1).
2. Продемонстрируйте навыки работы по загрузке модулей ядра с параметрами (см. раздел 10.4.2).

3 Выполнение лабораторной работы

10.4.1. Управление модулями ядра из командной строки

Запустим терминал в Rocky Linux от имени администратора (суперпользователя) через команду и введём пароль от учётной записи:

```
su -
```

Отобразим информацию об имеющихся устройствах в системе и связанными модулями (рис. 3.1):

```
lspci -k
```

Получим подробный список о контроллерах устройства (Ethernet-, USB-, аудио-контроллер и т.д.), а также связанные с ними драйвера или модули ядра, обозначенные *Kernel modules*. У контроллеров указаны названия (с указанием, в данном случае, компании Intel), у модулей и драйверов - названия.

```
[aeakunaeva@aeakunaeva ~]$ su -
Password:
[root@aeakunaeva ~]# lspci -k
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
    Kernel driver in use: ata_piix
    Kernel modules: ata_piix, ata_generic
00:02.0 VGA compatible controller: VMware SVGA II Adapter
    Subsystem: VMware SVGA II Adapter
    Kernel driver in use: vmwgfx
    Kernel modules: vmwgfx
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
    Subsystem: Intel Corporation PRO/1000 MT Desktop Adapter
    Kernel driver in use: e1000
    Kernel modules: e1000
00:04.0 System peripheral: InnoTek Systemberatung GmbH VirtualBox Guest Service
00:05.0 Multimedia audio controller: Intel Corporation 82801AA AC'97 Audio Controller (rev 01)
    Subsystem: Dell Device 0177
    Kernel driver in use: snd_intel8x0
    Kernel modules: snd_intel8x0
00:06.0 USB controller: Apple Inc. KeyLargo/Intrepid USB
    Kernel driver in use: ohci-pci
00:07.0 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
    Kernel driver in use: pix4_smbus
    Kernel modules: i2c_piix4
00:08.0 USB controller: Intel Corporation 82801FB/FBM/FR/FRW (ICH6 Family) USB2 EHCI Controller
    Kernel driver in use: ehci-pci
00:0d.0 SATA controller: Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E) SATA Controller [AHCI mode]
(rev 02)
    Kernel driver in use: ahci
    Kernel modules: ahci
```

Рис. 3.1: Отображение информации об устройствах системы и модулях

Отобразим список загруженных модулей ядра через утилиту *lsmod*, через pipe также добавив команду *sort* для сортировки модулей по названию в алфавитном порядке для удобства (рис. 3.2):

```
lsmod | sort
```

```
[root@aeakunaeva ~]# lsmod | sort
ac97_bus           12288  1 snd_ac97_codec
ahci                49152  3
ata_generic        16384  0
ata_piix            45056  0
cdrom              90112  1 sr_mod
crc32c_intel       24576  1
crc32_pclmul       12288  0
crc32c_pclmul      12288  1
dm_log              24576  2 dm_region_hash,dm_mirror
dm_mirror            28672  0
dm_mod              245760  9 dm_log,dm_mirror
dm_region_hash       28672  1 dm_mirror
drm                 811008  6 vmwgfx,drm_kms_helper,drm_ttm_helper,ttm
drm_kms_helper     266240  2 vmwgfx,drm_ttm_helper
drm_ttm_helper       16384  2 vmwgfx
e1000               196608  0
fuse                212992  5
ghash_clmulni_intel 16384  0
i2c_piix4           28672  0
intel_rapl_common   57344  1 intel_rapl_msr
intel_rapl_msr       20480  0
ip_set                69632  0
libahci              61440  1 ahci
libata               520192  4 ata_piix,libahci,ahci,ata_generic
libcrc32c            12288  4 nf_conntrack,nf_nat,nf_tables,xfs
Module             Size Used by
nf_conntrack         229376  2 nf_nat,nft_ct
nf_defrag_ipv4        12288  1 nf_conntrack
nf_defrag_ipv6        24576  1 nf_conntrack
nf_nat                65536  1 nft_chain_nat
nfnetlink            20480  3 nf_tables,ip_set
```

Рис. 3.2: Отображение списка загруженных модулей ядра

Проверим, загружен ли модуль ядра ext4, через команду *lsmod* пытаясь найти его с указанием *grep* и названия модуля через pipe (рис. 3.3):

```
lsmod | grep ext4
```

Мы не получим никакой информации, т.к. модуль ext4 не загружен. Тогда загрузим его через следующую утилиту *modprobe* для загрузки модулей ядра и проверим, был ли модуль загружен, тем же способом:

```
modprobe ext4
```

```
lsmod | grep ext4
```

На этот раз мы получим информацию о модуле, что будет значить, что он активен. Выведем информацию о модуле ядра через утилиту *modinfo*:

```
modinfo ext4
```

Здесь мы увидим полную информацию о модуле: название, лицензия, подпись, описание и ключи модуля (предназначенные для Rocky), тем не менее не будут указаны параметры, т.к. их нет.

```
[root@aakunaeva ~]# lsmod | grep ext4
[root@aakunaeva ~]# modprobe ext4
[root@aakunaeva ~]# lsmod | grep ext4
ext4          1191936  0
mbcache        16384  1 ext4
jbd2         221184  1 ext4
[root@aakunaeva ~]# modinfo ext4
filename:      /lib/modules/5.14.0-570.17.1.el9_6.x86_64/kernel/fs/ext4/ext4.ko.xz
softdep:       pre: crc32c
license:       GPL
description:   Fourth Extended Filesystem
author:        Remy Card, Stephen Tweedie, Andrew Morton, Andreas Dilger, Theodore Ts'o and others
alias:         fs-ext4
alias:         ext3
alias:         fs-ext3
alias:         ext2
alias:         fs-ext2
rhelversion:   9.6
srcversion:    414CA771338855FD8AA67C4
depends:       jbd2,mbcache
retpoline:     Y
intree:        Y
name:          ext4
vermagic:     5.14.0-570.17.1.el9_6.x86_64 SMP preempt mod_unload modversions
sig_id:        PKCS#7
signer:        Rocky kernel signing key
sig_key:       F5:03:24:D1:25:4A:DE:82:57:F2:1C:EE:7C:D6:C7:14:28:E3:FF
sig_hashalgo:  sha256
signature:    61:BC:53:9F:C0:C1:2A:2D:B2:34:DF:10:67:4E:50:C6:10:A5:13:AB:
              2E:2B:4D:A7:EE:AD:A3:55:E7:39:D4:25:67:53:88:1E:0B:8F:0F:05:
              90:48:2A:CA:64:CD:DF:62:5D:18:2B:9A:B2:D2:D1:6A:9B:18:DF:60:
              8B:12:FF:F0:DC:FB:1B:10:EE:F8:3C:FB:3E:BC:58:4C:5F:12:47:C3:
```

Рис. 3.3: Проверка загрузки модуля ядра (ext4)

Теперь попробуем выгрузить модуль ext4 той же командой, но с добавлением ключа -r (рис. 3.4):

```
modprobe -r ext4
```

Получим сперва фатальную ошибку, т.к. модуль “используется”. Повторим выгрузку и на этот раз успешно (проверим дополнительно через lsmod | grep ext4):

```
modprobe -r ext4
```

```
lsmod | grep ext4
```

Выгрузим также модуль ядра xfs: однако, обнаружим, что не все модули ядра могут быть отключены повторным вводом команды (получим фатальную ошибку, т.к. модуль используется ядром):

```
modprobe -r xfs
```

```
[root@aeakunaeva ~]# modprobe -r ext4
modprobe: FATAL: Module crc32c_intel is in use.
[root@aeakunaeva ~]# modprobe -r ext4
[root@aeakunaeva ~]# lsmod | grep ext4
[root@aeakunaeva ~]# modprobe -r xfs
modprobe: FATAL: Module xfs is in use.
[root@aeakunaeva ~]# modprobe -r xfs
modprobe: FATAL: Module xfs is in use.
```

Рис. 3.4: Выгрузка модулей ядра (ext4 и xfs)

10.4.2. Загрузка модулей ядра с параметрами

Также, через суперпользователя, проверим, загружен ли модуль ядра bluetooth (отрицательно) (рис. 3.5):

```
lsmod | grep bluetooth
```

Тогда загрузим его и проверим ещё раз, загружен ли модуль, заодно выведем список модулей ядра, отвечающих за работу с Bluetooth:

```
modprobe bluetooth  
lsmod | grep bluetooth
```

Затем выведем информацию о модуле bluetooth: получим такой же набор с названием, описанием, лицензией, подписью и т.п., только в конце будут указаны три параметра, которые могут быть установлены для работы модуля (рис. 3.6):

1. disable_esc0

2. disable_ertm

3. enable_ecred

```
modinfo bluetooth
```

Выгрузим модуль:

```
modprobe -r bluetooth
```

```
[root@aekunaeva ~]# lsmod | grep bluetooth  
[root@aekunaeva ~]# modprobe bluetooth  
[root@aekunaeva ~]# lsmod | grep bluetooth  
bluetooth           1114112  0  
rfkill                  40960  4 bluetooth  
[root@aekunaeva ~]# modinfo blurtooth  
modinfo: ERROR: Module blurtooth not found.  
[root@aekunaeva ~]# modinfo bluetooth  
filename:      /lib/modules/5.14.0-570.17.1.el9_6.x86_64/kernel/net/bluetooth/bluetooth.ko.xz  
alias:         net-pf-31  
license:       GPL  
version:       2.22  
description:  Bluetooth Core ver 2.22  
author:        Marcel Holtmann <marcel@holtmann.org>  
rhelversion:   9.6  
srcversion:    7FC17935133A8AA3F10AD0C  
depends:       rfkill  
retpoline:     Y  
intree:        Y  
name:          bluetooth  
vermagic:     5.14.0-570.17.1.el9_6.x86_64 SMP preempt mod_unload modversions  
sig_id:        PKCS#7  
signer:        Rocky kernel signing key  
sig_key:       F5:03:24:D1:25:4A:DE:82:57:F2:1C:EE:7C:D6:C7:14:28:E3:FF  
sig_hashalgo:  sha256  
signature:    60:45:4D:31:29:07:01:AA:4A:4B:CF:C6:7C:F0:40:01:07:FA:B9:60:  
53:60:AC:A4:F8:DE:0F:4A:44:BC:57:6B:F6:6C:E5:4F:AF:37:F7:96:  
F7:A3:99:62:88:DB:5C:34:35:06:4C:97:14:79:17:96:FA:71:CE:F0:  
99:3D:32:04:E9:C3:61:60:35:DA:47:8C:CB:D0:40:5C:4E:A2:F4:68:  
C9:5E:A7:48:64:BA:24:71:C5:44:CF:60:4F:1D:35:F6:73:77:27:F9:  
07:5A:A2:60:E7:FE:AB:D3:54:F1:93:2F:24:37:CA:DA:CF:3C:9B:6A:  
26:96:D7:C9:C9:5D:91:E4:1E:7B:5B:F3:66:EA:9B:47:B4:4A:2A:E4:
```

Рис. 3.5: Загрузка модуля ядра bluetooth

```
9A:32:6B:B9:7B:D4:F1:3A:8A:BA:A8:6E:E4:7F:EE:9A:60:C6:AE:A5:  
8D:8D:23:7F:E3:2F:EF:4B:4C:43:3E:52:F8:9C:B1:55:15:77:66:DF:  
42:04:B9:4E:1D:9D:6C:3B:4B:80:C3:43:3F:5A:58:AE:DB:BB:E5:82:  
D7:76:76:34:6C:F2:0F:69:38:03:DF:A4:89:A7:30:61:8D:D5:07:72:  
2C:3E:9E:A3:35:EE:98:8D:64:AD:07:07:DA:73:75:20:FE:44:28:BC:  
7C:A3:26:F7:82:B8:DA:B1:0A:8B:87:F2:69:2D:F8:F5:7F:0C:2B:C2:  
B3:40:73:8F:1D:48:EE:62:F7:08:43:33:A1:18:52:7E:F3:91:E1:49:  
A7:61:FF:DA:FA:6C:F5:DD:B3:AB:B2:9D:AA:DB:58:41:CF:F8:38:0B:  
EE:C1:85:7E:14:6C:AF:33:F8:40:B4:1B:BD:0B:8A:53:5D:6C:EE:84:  
47:1E:DA:9F  
parm: disable_esco:Disable eSCO connection creation (bool)  
parm: disable_erm:Disable enhanced retransmission mode (bool)  
parm: enable_ecred:Enable enhanced credit flow control mode (bool)  
[root@aeakunaeva ~]# modprobe -r bluetooth  
[root@aeakunaeva ~]# lsmod | grep bluetooth
```

Рис. 3.6: Параметры bluetooth и выгрузка модуля

10.4.3. Обновление ядра системы

Как суперпользователь, определим текущую версию ядра ОС (5.14.0-570.17.1.el9_6.86_64) (рис. 3.7):

```
uname -r
```

И выведем список пакетов, которые относятся к ядру: в нашем случае это установленный пакет kernel.x86_64 5.14.0-570.17.1.el9_6.86_64, но также отображён будет доступный для установки пакет новой версии ядра 5.14.0-570.42.2.el9_6.86_64:

```
dnf list kernel
```

```
[root@aeakunaeva ~]# uname -r  
5.14.0-570.17.1.el9_6.x86_64  
[root@aeakunaeva ~]# dnf list kernel  
Last metadata expiration check: 0:48:41 ago on Sat 04 Oct 2025 09:50:08 PM MSK.  
Installed Packages  
kernel.x86_64          5.14.0-570.17.1.el9_6                      @anaconda  
Available Packages  
kernel.x86_64          5.14.0-570.42.2.el9_6                      baseos
```

Рис. 3.7: Информация о текущей версии ядра и доступных пакетах

Обновим систему во избежание конфликтов между пакетами ядра (рис. 3.8):

```
dnf upgrade --refresh
```

```
[root@aekunaeva ~]# dnf upgrade --refresh
Rocky Linux 9 - BaseOS
Rocky Linux 9 - AppStream
Rocky Linux 9 - Extras
Dependencies resolved.
=====
           Package          Arch    Version        Repository      Size
=====
Installing:
  kernel            x86_64  5.14.0-570.58.1.el9_6      baseos     1.8 M
Upgrading:
  NetworkManager    x86_64  1:1.52.0-9.el9_6      baseos     2.3 M
  NetworkManager-adsl x86_64  1:1.52.0-9.el9_6      baseos     31 k
  NetworkManager-bluetooth x86_64  1:1.52.0-9.el9_6      baseos     57 k
  NetworkManager-config-server noarch  1:1.52.0-9.el9_6      baseos     17 k
  NetworkManager-libmm x86_64  1:1.52.0-9.el9_6      baseos     1.9 M
  NetworkManager-team x86_64  1:1.52.0-9.el9_6      baseos     36 k
  NetworkManager-tui  x86_64  1:1.52.0-9.el9_6      baseos     246 k
  NetworkManager-wifi x86_64  1:1.52.0-9.el9_6      baseos     80 k
  NetworkManager-wwan x86_64  1:1.52.0-9.el9_6      baseos     64 k
  at                x86_64  3.1.23-12.el9_6      baseos     59 k
  avahi             x86_64  0.8-22.el9_6.1      baseos     287 k
  avahi-glib         x86_64  0.8-22.el9_6.1      appstream   14 k
  avahi-libs          x86_64  0.8-22.el9_6.1      baseos     66 k
  avahi-tools         x86_64  0.8-22.el9_6.1      appstream   38 k
  bind-libs           x86_64  32:9.16.23-31.el9_6    appstream   1.2 M
  bind-license        noarch  32:9.16.23-31.el9_6    appstream   12 k
  bind-utils           x86_64  32:9.16.23-31.el9_6    appstream   199 k
  buildah            x86_64  2:11.39.4-2.el9_6      appstream   11 M
  cockpit            x86_64  224.7-1.el9_6      baseos     40 k
=====

```

Рис. 3.8: Обновление системы

После обновления системы обновим ядро ОС и саму ОС после: будут установлены новые версии из доступных пакетов (рис. 3.9):

```
dnf update kernel
dnf update
dnf upgrade --refresh
```

```
Installed:
  grub2-tools-efi-1:2.06-105.el9_6.x86_64      grub2-tools-extra-1:2.06-105.el9_6.x86_64
  kernel-5.14.0-570.58.1.el9_6.x86_64      kernel-core-5.14.0-570.58.1.el9_6.x86_64
  kernel-devel-5.14.0-570.58.1.el9_6.x86_64    kernel-modules-5.14.0-570.58.1.el9_6.x86_64
  kernel-modules-core-5.14.0-570.58.1.el9_6.x86_64

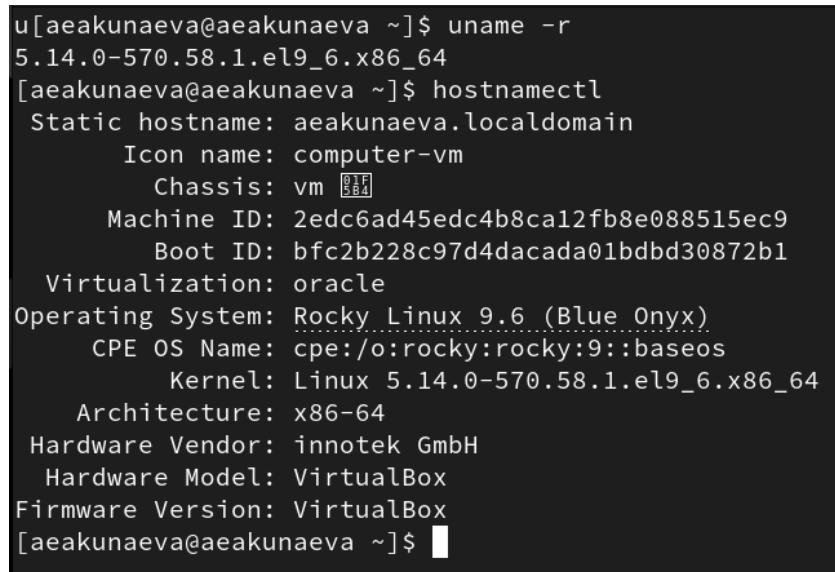
Complete!
[root@aekunaeva ~]# dnf update kernel
Last metadata expiration check: 0:12:45 ago on Sat 04 Oct 2025 10:44:03 PM MSK.
Dependencies resolved.
Nothing to do.
Complete!
[root@aekunaeva ~]# dnf update kernel
Rocky Linux 9 - BaseOS
Rocky Linux 9 - AppStream
Rocky Linux 9 - Extras
Dependencies resolved.
Nothing to do.
Complete!
[root@aekunaeva ~]# dnf update
Last metadata expiration check: 0:00:24 ago on Sat 08 Nov 2025 04:29:36 AM MSK.
Dependencies resolved.
Nothing to do.
Complete!
[root@aekunaeva ~]# dnf upgrade --refresh
Rocky Linux 9 - BaseOS
Rocky Linux 9 - AppStream
Rocky Linux 9 - Extras
Dependencies resolved.
Nothing to do.
Complete!
[root@aekunaeva ~]#
```

Рис. 3.9: Обновление ядра и ОС

Затем перезагрузим систему: появится подпись *Blue Onyx* при выборе ядра. Выбираем первое и самое доступное, заходим под своей учётной записью и проверяем версию ядра в терминале (новая - 5.14.0-570.58.1.el9_6.86_64) (рис. 3.10):

```
uname -r  
hostnamectl
```

Теперь установлена новая версия ядра ОС и при получении более подробной информации о ядре, также убеждаемся в этом в строке *Kernel*.



```
u[aeakunaeva@aeakunaeva ~]$ uname -r  
5.14.0-570.58.1.el9_6.x86_64  
[aeakunaeva@aeakunaeva ~]$ hostnamectl  
  Static hostname: aeakunaeva.localdomain  
    Icon name: computer-vm  
      Chassis: vm [0TF]  
    Machine ID: 2edc6ad45edc4b8ca12fb8e088515ec9  
      Boot ID: bfc2b228c97d4dacada01bdbd30872b1  
Virtualization: oracle  
Operating System: Rocky Linux 9.6 (Blue Onyx)  
  CPE OS Name: cpe:/o:rocky:rocky:9::baseos  
    Kernel: Linux 5.14.0-570.58.1.el9_6.x86_64  
  Architecture: x86-64  
Hardware Vendor: innotek GmbH  
  Hardware Model: VirtualBox  
Firmware Version: VirtualBox  
[aeakunaeva@aeakunaeva ~]$ █
```

Рис. 3.10: Проверка установки новой версии ядра ОС

4 Контрольные вопросы

1. Какая команда показывает текущую версию ядра, которая используется на вашей системе?

```
uname -r
```

``` ([рис. @fig:011]):

! [Контрольный вопрос №1] (image/q1.PNG){#fig:011 width=70%}

\*\*2. Как можно посмотреть более подробную информацию о текущей версии ядра операционной системы?

hostnamectl “ (рис. 4.1):

```
[aeakunaeva@aeakunaeva ~]$ hostnamectl
Static hostname: aeakunaeva.localdomain
Icon name: computer-vm
Chassis: vm [0.1]
Machine ID: 2edc6ad45edc4b8ca12fb8e088515ec9
Boot ID: bfc2b228c97d4dacada01bdbd30872b1
Virtualization: oracle
Operating System: Rocky Linux 9.6 (Blue Onyx)
CPE OS Name: cpe:/o:rocky:rocky:9::baseos
Kernel: Linux 5.14.0-570.58.1.el9_6.x86_64
Architecture: x86-64
Hardware Vendor: innotek GmbH
Hardware Model: VirtualBox
Firmware Version: VirtualBox
```

Рис. 4.1: Контрольный вопрос №2

**3. Какая команда показывает список загруженных модулей ядра?**

```
lsmod
```

(Пояснение: на рис. добавлен параметр sort через pipe для удобного отображения информации) (рис. 4.2):

```
[root@aeakunaeva ~]# lsmod | sort
ac97_bus 12288 1 snd_ac97_codec
ahci 49152 3
ata_generic 16384 0
ata_piix 45056 0
cdrom 90112 1 sr_mod
crc32c_intel 24576 1
crc32_pclmul 12288 0
crct10dif_pclmul 12288 1
dm_log 24576 2 dm_region_hash,dm_mirror
dm_mirror 28672 0
dm_mod 245760 9 dm_log,dm_mirror
dm_region_hash 28672 1 dm_mirror
drm 811008 6 vmwgfx,drm_kms_helper,drm_ttm_helper,ttm
drm_kms_helper 266240 2 vmwgfx,drm_ttm_helper
drm_ttm_helper 16384 2 vmwgfx
e1000 196608 0
fuse 212992 5
ghash_clmulni_intel 16384 0
i2c_piix4 28672 0
```

Рис. 4.2: Контрольный вопрос №3

#### 4. Какая команда позволяет вам определять параметры модуля ядра?

Команда modinfo позволяет получить информацию о модуле ядра, в том числе об имеющихся параметрах (в самом конце выведенного списка) (рис. 4.3-рис. 4.4):

modinfo [модуль ядра]

```
[root@aeakunaeva ~]# modinfo bluetooth
filename: /lib/modules/5.14.0-570.17.1.e19_6.x86_64/kernel/net/bluetooth/bluetooth.ko.xz
alias: net-pf-31
license: GPL
version: 2.22
description: Bluetooth Core ver 2.22
author: Marcel Holtmann <marcel@holtmann.org>
rhelversion: 9.6
srcversion: 7FC17935133A8AA3F10AD0C
depends: rfkill
retpoline: Y
intree: Y
name: bluetooth
vermagic: 5.14.0-570.17.1.e19_6.x86_64 SMP preempt mod_unload modversions
sig_id: PKCS#7
signer: Rocky kernel signing key
sig_key: F5:03:24:D1:25:4A:DE:82:57:F2:1C:EE:7C:D6:C7:14:28:E3:FF
sig_hashalgo: sha256
signature: 60:45:4D:31:29:07:01:AA:4A:4B:CF:C6:7C:F0:40:01:07:FA:B9:60:
 53:60:AC:A4:F8:DE:0F:4A:44:BC:57:6B:F6:6C:E5:4F:AF:37:F7:96:
 F7:A3:99:62:88:DB:5C:34:35:06:4C:97:14:79:17:96:FA:71:CE:F0:
```

Рис. 4.3: Контрольный вопрос №4-1

```
B3:40:73:8F:1D:48:EE:62:F7:08:43:33:A1:18:52:7E:F3:91:E1:49:
A7:61:FF:DA:FA:6C:F5:DD:B3:AB:B2:9D:AA:DB:58:41:CF:F8:38:0B:
EE:C1:85:7E:14:6C:AF:33:F8:40:B4:1B:BD:0B:8A:53:5D:6C:EE:84:
47:1E:DA:9F
parm: disable_esco:Disable eSCO connection creation (bool)
parm: disable_erm:Disable enhanced retransmission mode (bool)
parm: enable_ecred:Enable enhanced credit flow control mode (bool)
```

Рис. 4.4: Контрольный вопрос №4-2

## 5. Как выгрузить модуль ядра?

Через утилиту modprobe с указанием параметра -r (проверить загрузку/выгрузку можно через lsmod | grep [модуль ядра]) (рис. 4.5):

```
modprobe -r [модуль ядра]
```

```
[root@aeakunaeva ~]# modprobe -r ext4
[root@aeakunaeva ~]# lsmod | grep ext4
```

Рис. 4.5: Контрольный вопрос №5

## 6. Что вы можете сделать, если получите сообщение об ошибке при попытке выгрузить модуль ядра?

Можно попытаться повторно ввести команду для выгрузки. Однако, это неизменно поможет (рис. 4.6):

```
[root@aeakunaeva ~]# modprobe -r ext4
modprobe: FATAL: Module crc32c_intel is in use.
[root@aeakunaeva ~]# modprobe -r ext4
[root@aeakunaeva ~]# lsmod | grep ext4
```

Рис. 4.6: Контрольный вопрос №6

## 7. Как определить, какие параметры модуля ядра поддерживаются?

Через modinfo можно получить информацию о поддерживаемых модулях ядра (рис. 4.7):

```
modinfo [модуль ядра]
```

```
parm: disable_esco:Disable eSCO connection creation (bool)
parm: disable_erm:Disable enhanced retransmission mode (bool)
parm: enable_ecred:Enable enhanced credit flow control mode (bool)
```

Рис. 4.7: Контрольный вопрос №7

## 8. Как установить новую версию ядра?

С предварительным обновлением системы во избежание непредвиденных ситуаций, можно обновить ядро следующим набором команд и после перезагрузить систему (рис. 4.8):

```
dnf upgrade --refresh
dnf update kernel
dnf update
dnf upgrade --refresh
```

```
[root@aekunaeva ~]# dnf update kernel
Rocky Linux 9 - BaseOS 12 kB/s | 4.1 kB 00:00
Rocky Linux 9 - AppStream 9.9 kB/s | 4.5 kB 00:00
Rocky Linux 9 - Extras 9.1 kB/s | 2.9 kB 00:00
Dependencies resolved.
Nothing to do.
Complete!
[root@aekunaeva ~]# dnf update
Last metadata expiration check: 0:00:24 ago on Sat 08 Nov 2025 04:29:36 AM MSK.
Dependencies resolved.
Nothing to do.
Complete!
[root@aekunaeva ~]# dnf upgrade --refresh
Rocky Linux 9 - BaseOS 13 kB/s | 4.1 kB 00:00
Rocky Linux 9 - AppStream 13 kB/s | 4.5 kB 00:00
Rocky Linux 9 - Extras 9.0 kB/s | 2.9 kB 00:00
Dependencies resolved.
Nothing to do.
Complete!
[root@aekunaeva ~]#
```

Рис. 4.8: Контрольный вопрос №8

## **5 Выводы**

Я получила навыки работы с утилитами управления модулями ядра операционной системы.

## **Список литературы**

1. Кулабов Д.С. Основы работы с модулями ядра операционной системы. ТУИС РУДН.