

# **Отчёт по лабораторной работе №10**

**Основы работы с модулями ядра операционной системы**

**Щемелев Илья Владимирович**

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Ход выполнения</b>	<b>6</b>
2.1 Управление модулями ядра из командной строки . . . . .	6
2.2 Работа с модулем файловой системы ext4 . . . . .	9
2.3 Загрузка модулей ядра с параметрами (на примере bluetooth) . . . . .	12
2.4 Обновление ядра системы . . . . .	13
<b>3 Контрольные вопросы</b>	<b>16</b>
<b>4 Заключение</b>	<b>19</b>

# Список иллюстраций

2.1	Просмотр устройств и связанных модулей (lspci -k) . . . . .	8
2.2	Список загруженных модулей (lsmod   sort) . . . . .	9
2.3	Загрузка ext4 и проверка наличия в lsmod . . . . .	10
2.4	Попытка выгрузки ext4 и ошибка при выгрузке xfs . . . . .	11
2.5	Загрузка bluetooth и просмотр связанных модулей . . . . .	12
2.6	Параметры bluetooth и выгрузка модуля . . . . .	13
2.7	Проверка версии ядра и попытка обновления через dnf . . . . .	14
2.8	Проверка версии ядра и сведений о системе (hostnamectl) . . . . .	15

# **Список таблиц**

# **1 Цель работы**

Получить навыки работы с утилитами управления модулями ядра операционной системы.

## 2 Ход выполнения

### 2.1 Управление модулями ядра из командной строки

1. В системе запущен терминал и получены полномочия администратора (пользователь **root**) командой **su -**.

Это необходимо, поскольку просмотр информации о драйверах/модулях и операции **modprobe** (загрузка/выгрузка модулей) требуют прав суперпользователя.

2. Для определения доступных устройств и связанных с ними модулей ядра выполнена команда **lspci -k**.

В выводе отображаются PCI-устройства, а также строки **Kernel driver in use** и **Kernel modules**, позволяющие установить, какой модуль обслуживает конкретное устройство и какие модули доступны для него.

По результатам команды были выявлены основные устройства виртуальной машины и соответствующие им модули:

- **IDE interface (Intel PIIX4 IDE)** – используется драйвер **ata\_piix**, дополнительно доступен **ata\_generic**.

Это типичная подсистема хранения для виртуализированных конфигураций с PIIX-чипсетом.

- **VGA compatible controller (VMware SVGA II Adapter)** – используется драйвер **vmwgfx**.

Такой адаптер часто встречается в виртуальных средах; модуль обес-

печивает работу графической подсистемы.

- **Ethernet controller (Intel 82540EM Gigabit Ethernet Controller)** — используется драйвер **e1000**.

Данный сетевой адаптер является распространённым «виртуальным» устройством, поддерживаемым модулем e1000.

- **VirtualBox Guest Service (InnoTek Systemberatung GmbH)** — используется модуль **vboxguest**.

Он относится к гостевым дополнениям VirtualBox и обеспечивает интеграцию гостевой ОС с хостом.

- **Multimedia audio controller (Intel AC'97)** — используется драйвер **snd\_intel8x0** (а также связанные аудиомодули ALSA).
- **SATA controller (Intel ICH8M/ICH9M-E SATA AHCI)** — используется драйвер **ahci**, обеспечивающий работу SATA-контроллера в режиме AHCI.

```

root@ivschemelev:/home/ivschemelev# lspci -k
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIIX4 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
    Kernel driver in use: vboxguest
    Kernel modules: vboxguest
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 PIIIX4 ACPI (rev 08)
    Kernel driver in use: piix4_smbus
    Kernel modules: i2c_piix4
00:0b.0 USB controller: Intel Corporation 82801FB/FBM/FR/fw/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
root@ivschemelev:/home/ivschemelev# 

```

Рис. 2.1: Просмотр устройств и связанных модулей (lspci -k)

3. Для просмотра списка загруженных модулей ядра выполнена команда `lsmod | sort`.

Вывод содержит таблицу с ключевыми полями:

- **Module** – имя модуля;
- **Size** – размер модуля в байтах;
- **Used by** – количество и список модулей/подсистем, которые используют данный модуль.

В списке присутствуют, в том числе, модули хранения (**ahci**, **ata\_piix**, **libata**), виртуализации/графики (**vmwgfx**), сети (**e1000**), а также системные компоненты (например, подсистема device-mapper – **dm\_mod**, поддержка FUSE – **fuse**).

```

root@ivschemelev:/home/ivschemelev#
root@ivschemelev:/home/ivschemelev# lsmod | sort
ac97_bus           12288  1 snd_ac97_codec
ahci              57344  3
ata_generic       12288  0
ata_piix          45056  1
cdrom             90112  2 isofs,sr_mod
crc32c_intel     12288  0
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_multipath,dm_log,dm_mirror
dm_multipath      53248  0
dm_region_hash    28672  1 dm_mirror
drm_ttm_helper    16384  2 vmwgfx
e1000             200704 0
fuse              253952 5
ghash_clmulni_intel 16384  0
i2c_piix4        36864  0
i2c_smbus         20480  1 i2c_piix4
intel_pmc_core   139264  0
intel_rapl_common 53248  1 intel_rapl_msrm
intel_rapl_msrm  20480  0
intel_uncore_frequency_common 16384  0
intel_vsec        20480  1 intel_pmc_core
iscsi_tcp         28672  0
isofs              69632  1
joydev             28672  0
libahci            69632  1 ahci
libata             512000 4 ata_piix,libahci,ahci,ata_generic
libiscsi           94208  2 libiscsi_tcp,iscsi_tcp
libiscsi_tcp      45056  1 iscsi_tcp
loop              45056  0
Module             Size  Used by

```

Рис. 2.2: Список загруженных модулей (lsmod | sort)

## 2.2 Работа с модулем файловой системы ext4

4. Проверено, загружен ли модуль **ext4**, командой `lsmod | grep ext4`.

На данном этапе совпадений не обнаружено, что означает отсутствие загруженного модуля **ext4** в памяти ядра на момент проверки.

5. Выполнена загрузка модуля **ext4** командой `modprobe ext4`.

Затем повторно выполнена проверка `lsmod | grep ext4`.

После загрузки в выводе появился модуль **ext4**, а также связанные зависи-

мости **jbd2** и **mbcache**, которые используются файловой системой ext4 для журналирования и кэширования метаданных.

```
root@ivschemelev:/home/ivschemelev#
root@ivschemelev:/home/ivschemelev# lsmod | grep ext4
root@ivschemelev:/home/ivschemelev# modprobe ext4
root@ivschemelev:/home/ivschemelev# lsmod | grep ext4
ext4          1187840  0
mbcache          16384  1 ext4
jbd2            217088  1 ext4
root@ivschemelev:/home/ivschemelev# modinfo ext4
filename:       /lib/modules/6.12.0-124.21.1.el10_1.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:    10.1
srcversion:     072609EF822F641F8FB2FBB
depends:        jbd2,mbcache
intree:         Y
name:           ext4
retpoline:      Y
vermagic:      6.12.0-124.21.1.el10_1.x86_64 SMP preempt mod_unload modversions
sig_id:         PKCS#7
signer:         Rocky kernel signing key
sig_key:        37:B8:CF:EC:A9:44:BE:6D:CB:5A:FD:C9:A1:0F:2E:AF:A2:5C:42:12

```

Рис. 2.3: Загрузка ext4 и проверка наличия в lsmod

## 6. Получена подробная информация о модуле ext4 командой `modinfo ext4`.

По результатам вывода можно сделать следующие пояснения по основным полям:

- **filename** — путь к файлу модуля в системе (расположение `.ko.xz` внутри каталога `/lib/modules/<версия_ядра>/...`);
- **license** — лицензия модуля (в данном случае **GPL**);
- **description** — назначение (файловая система Fourth Extended Filesystem);
- **author** — авторы/разработчики;
- **alias** — альтернативные идентификаторы, по которым модуль может автоматически подбираться системой (в т.ч. совместимость с **ext2/ext3**);

- **depends** — зависимости (например, **jbd2**, **mbcache**);
- **intree: Y** — модуль входит в состав основного дерева исходников ядра;
- **vermagic** — строка совместимости с конкретной версией/конфигурацией ядра (важно для корректной загрузки);
- **signer / sig\_key** — сведения о подписи модуля (используется при проверке целостности/подлинности).

Также следует отметить, что у модуля **ext4** отсутствуют настраиваемые параметры загрузки: в выводе нет строк вида **parm:** — это означает, что модуль не предоставляет пользовательских параметров через **modprobe**.

#### 7. Выполнена попытка выгрузки модуля **ext4** командой **modprobe -r ext4**.

Команда была введена несколько раз. Сообщений об ошибках система не вывела, что означает успешную выгрузку модуля при условии отсутствия активных точек монтирования ext4/зависимостей, препятствующих выгрузке.

#### 8. Выполнена попытка выгрузки модуля **xfs** командой **modprobe -r xfs**.

Система вернула ошибку **modprobe: FATAL: Module xfs is in use.**

Это означает, что модуль **xfs** задействован в текущей работе системы (как правило, файловая система XFS используется для одного из смонтированных разделов, часто — для корневого или системных томов), поэтому ядро запрещает его выгрузку во избежание повреждения данных и нарушения работы ОС.

```
root@ivschemelev:/home/ivschemelev#
root@ivschemelev:/home/ivschemelev# modprobe -r ext4
root@ivschemelev:/home/ivschemelev# modprobe -r ext4
root@ivschemelev:/home/ivschemelev# modprobe -r xfs
modprobe: FATAL: Module xfs is in use.
root@ivschemelev:/home/ivschemelev#
```

Рис. 2.4: Попытка выгрузки ext4 и ошибка при выгрузке xfs

## 2.3 Загрузка модулей ядра с параметрами (на примере bluetooth)

9. Проверено, загружен ли модуль **bluetooth**, командой `lsmod | grep bluetooth`.

До загрузки совпадений не обнаружено, следовательно, модуль Bluetooth не был активен.

10. Выполнена загрузка модуля Bluetooth командой `modprobe bluetooth`.

После чего повторно просмотрены связанные модули командой `lsmod | grep bluetooth`.

В выводе появился модуль **bluetooth**, а также модуль **rfkill**, который используется для программной блокировки/разблокировки радиоинтерфейсов (Bluetooth/Wi-Fi и др.) и является зависимостью Bluetooth-подсистемы.

```
root@ivschemelev:/home/ivschemelev#
root@ivschemelev:/home/ivschemelev# lsmod | grep bluetooth
root@ivschemelev:/home/ivschemelev# modprobe bluetooth
root@ivschemelev:/home/ivschemelev# lsmod | grep bluetooth
bluetooth           1118208  0
rfkill                40960  4 bluetooth
root@ivschemelev:/home/ivschemelev# modinfo bluetooth
filename:      /lib/modules/6.12.0-124.21.1.el10_1.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: 10.1
srcversion:   A561ADF438212A49F631F83
depends:     rfkill
intree:      Y
name:        bluetooth
retpoline:   Y
vermagic:    6.12.0-124.21.1.el10_1.x86_64 SMP preempt mod_unload modversions
sig_id:      PKCS#7
signer:      Rocky kernel signing key
sig_key:     37:B8:CF:EC:A9:44:BE:6D:CB:5A:FD:C9:A1:0F:2E:AF:A2:5C:42:12
sig_hashalgo: sha256
signature:   9D:15:FC:F7:8B:9F:7F:51:B2:76:0E:96:9D:8F:F1:38:DF:7E:0D:EF:
               8B:A1:ED:25:60:32:CA:5F:CE:0F:AC:CC:9E:12:36:0D:53:EB:0D:E4:
               3F:F1:7D:AF:FA:D5:B2:C6:03:1D:94:2F:6F:45:B4:A3:11:AD:3A:FB:
```

Рис. 2.5: Загрузка bluetooth и просмотр связанных модулей

11. Получена информация о модуле Bluetooth командой `modinfo bluetooth`.

Помимо стандартных полей (путь к модулю, описание, зависимости, vermagic, сведения о подписи), в конце вывода присутствуют строки **parm:**, то есть параметры, которые могут задаваться при загрузке модуля.

Для данного модуля доступны следующие параметры (все типа **bool**):

- **disable\_esco** — отключение создания eSCO-соединений (режим Bluetooth-аудио);
- **disable\_ertm** — отключение Enhanced Retransmission Mode (ERTM) для L2CAP;
- **enable\_ecred** — включение Enhanced Credit Based Flow Control Mode.

Наличие этих параметров позволяет изменять поведение Bluetooth-подсистемы при загрузке модуля, например: `modprobe bluetooth disable_ertm=1` (пример принципа задания параметра).

12. Выполнена выгрузка модуля Bluetooth командой `modprobe -r bluetooth`.

Выгрузка прошла без ошибок, что указывает на отсутствие активных Bluetooth-подключений/зависимостей, удерживающих модуль в памяти.

```
E0:01:50:13:7D:F0:3E:CB:01:95:47:7F:03:A0:A3:46:15:17:4D:FF:  
4E:59:55:A6  
parm: disable_esco:Disable eSCO connection creation (bool)  
parm: disable_ertm:Disable enhanced retransmission mode (bool)  
parm: enable_ecred:Enable enhanced credit flow control mode (bool)  
root@ivschemelev:/home/ivschemelev#  
root@ivschemelev:/home/ivschemelev# modprobe -r bluetooth  
root@ivschemelev:/home/ivschemelev#
```

Рис. 2.6: Параметры `bluetooth` и выгрузка модуля

## 2.4 Обновление ядра системы

13. Проверена текущая версия ядра командой `uname -r`.

Установлено, что система использует ядро версии **6.12.0-124.21.1.el10\_1.x86\_64**.

14. Просмотрен список пакетов, относящихся к ядру, командой `dnf list kernel`.

В системе присутствуют пакеты ядра, включая установленное ядро текущей версии, а также пакет предыдущей версии, оставленный в системе для возможности отката (типичное поведение при обновлениях ядра).

15. Выполнены команды обновления `dnf update kernel`, затем `dnf update` и `dnf upgrade --refresh`.

По выводу видно, что зависимости разрешены успешно, однако обновления не требуются — система сообщает **Nothing to do**.

Это означает, что на момент проверки установлены актуальные пакеты (включая ядро), доступные в подключённых репозиториях.

```
root@ivschemelev:/home/ivschemelev#
root@ivschemelev:/home/ivschemelev# uname -r
6.12.0-124.21.1.el10_1.x86_64
root@ivschemelev:/home/ivschemelev# dnf list kernel
Last metadata expiration check: 2:42:00 ago on Fri 16 Jan 2026 11:25:39 AM MSK.
Installed Packages
kernel.x86_64                  6.12.0-55.12.1.el10_0          @anaconda
kernel.x86_64                  6.12.0-124.21.1.el10_1          @baseos
root@ivschemelev:/home/ivschemelev# dnf update kernel
Last metadata expiration check: 2:42:17 ago on Fri 16 Jan 2026 11:25:39 AM MSK.
Dependencies resolved.
Nothing to do.
Complete!
root@ivschemelev:/home/ivschemelev# dnf update
Last metadata expiration check: 2:42:21 ago on Fri 16 Jan 2026 11:25:39 AM MSK.
Dependencies resolved.
Nothing to do.
Complete!
root@ivschemelev:/home/ivschemelev# dnf upgrade --refresh
Rocky Linux 10 - BaseOS           7.5 kB/s | 4.3 kB   00:00
Rocky Linux 10 - AppStream         15 kB/s | 4.3 kB   00:00
Rocky Linux 10 - Extras           1.6 kB/s | 3.1 kB   00:01
Dependencies resolved.
Nothing to do.
Complete!
root@ivschemelev:/home/ivschemelev#
```

Рис. 2.7: Проверка версии ядра и попытка обновления через `dnf`

16. Выполнена перезагрузка системы (перезапуск виртуальной машины). После загрузки повторно проверены версия ядра и сведения о системе командами `uname -r` и `hostnamectl`.

Версия ядра после перезагрузки подтверждена как **6.12.0-124.21.1.el10\_1.x86\_64**.

Команда `hostnamectl` дополнительно показала параметры окружения:

- **Operating System:** Rocky Linux **10.1 (Red Quartz);**
- **Virtualization:** **oracle** (VirtualBox);
- сведения о хостнейме (**ivschemelev.localdomain**) и аппаратной модели (**VirtualBox**).

Это подтверждает, что система работает в среде виртуализации VirtualBox и использует актуальное ядро, установленное в ОС.

```

root@ivschemelev:/home/ivschemelev# uname -r
6.12.0-124.21.1.el10_1.x86_64
root@ivschemelev:/home/ivschemelev# hostnamectl
  Static hostname: ivschemelev.localdomain
    Icon name: computer-vm
      Chassis: vm 🖥
    Machine ID: 473c978a805e47e9bc9a702cdd313842
      Boot ID: 4dc896ba0b72485698134d5402aa29e8
    Product UUID: dd1cb896-afb2-384b-89c4-2b7603fd2490
  Virtualization: oracle
Operating System: Rocky Linux 10.1 (Red Quartz)
      CPE OS Name: cpe:/o:rocky:rocky:10::baseos
        OS Support End: Thu 2035-05-31
OS Support Remaining: 9y 4month 1w 5d
          Kernel: Linux 6.12.0-124.21.1.el10_1.x86_64
          Architecture: x86-64
        Hardware Vendor: innotek GmbH
        Hardware Model: VirtualBox
        Hardware Serial: VirtualBox-96b81cdd-b2af-4b38-89c4-2b7603fd2490
      Firmware Version: VirtualBox
        Firmware Date: Fri 2006-12-01
        Firmware Age: 19y 1month 2w 2d
root@ivschemelev:/home/ivschemelev#

```

---

Рис. 2.8: Проверка версии ядра и сведений о системе (hostnamectl)

## 3 Контрольные вопросы

1. Текущую версию ядра, которая используется на системе, показывает команда:
  - **uname -r** — выводит строку версии загруженного ядра (release). Данная команда удобна для быстрой проверки, какое именно ядро в данный момент активно.
2. Более подробную информацию о текущей версии ядра операционной системы можно получить следующими способами:
  - **uname -a** — выводит расширенные сведения: имя ядра, имя хоста, версию, дату сборки и архитектуру. Это наиболее полный вариант вывода среди команд `uname`.
  - **hostnamectl** — отображает сведения о системе, включая строку **Kernel**, а также информацию об ОС, архитектуре и среде виртуализации (если используется). Данный вариант удобен, когда нужно получить общую «паспортную» информацию о системе.
3. Список загруженных модулей ядра показывает команда:
  - **lsmod** — выводит перечень модулей, их размер и количество/список зависимостей (кто использует модуль). Часто применяется совместно с фильтрацией, например `lsmod | grep имя_модуля`.
4. Команда, позволяющая определять параметры модуля ядра:

- **modinfo имя\_модуля** — отображает сведения о модуле, включая поддерживаемые параметры (строки вида **parm:**). Если параметров нет, значит модуль не предоставляет настраиваемых опций через modprobe.

5. Для выгрузки модуля ядра используется команда:

- **modprobe -r имя\_модуля** — корректно выгружает модуль и при необходимости пытается выгрузить связанные зависимости. Дополнительно также существует:
- **rmmod имя\_модуля** — выгружает модуль напрямую, но обычно предпочтительнее применять modprobe -r, так как он учитывает зависимости.

6. Если при попытке выгрузить модуль ядра получено сообщение об ошибке, можно выполнить следующие действия:

- убедиться, что модуль действительно можно выгружать (часто он используется системой прямо сейчас);
- определить, какие процессы/подсистемы удерживают модуль (например, устройство активно или файловая система смонтирована);
- остановить службы или завершить процессы, связанные с использованием устройства/подсистемы (например, отключить сервис, закрыть соединения);
- размонтировать файловую систему, если модуль относится к используемой ФС (в этом случае выгрузка станет возможной);
- если модуль критически важен для работы ОС (например, драйвер корневой файловой системы), выгружать его нельзя — в таких ситуациях допускается только перезагрузка в другую конфигурацию либо изменение состава модулей через настройки загрузки/сборку initramfs.

7. Определить, какие параметры модуля ядра поддерживаются, можно так:

- **modinfo имя\_модуля** — параметры перечисляются в строках **parm:**

(имя параметра и его назначение). Например, для некоторых модулей можно увидеть параметры типа `parm: option=... (bool/int/string)`.

8. Установить новую версию ядра можно следующими действиями:

- обновить информацию о пакетах и выполнить обновление ядра пакетным менеджером:
  - **`dnf update kernel`**
- при необходимости обновить систему целиком, чтобы исключить конфликты зависимостей:
  - **`dnf upgrade --refresh`**
- перезагрузить систему и выбрать новое ядро в меню загрузчика (если доступен выбор);
- после загрузки проверить, что используется новое ядро:
  - **`uname -r`**
  - при необходимости дополнительно проверить через **`hostnamectl`** (строка Kernel).

## **4 Заключение**

В ходе выполнения лабораторной работы были изучены основные приёмы управления модулями ядра операционной системы Rocky Linux. Были рассмотрены способы получения информации об аппаратных устройствах и связанных с ними модулях, а также выполнены операции загрузки и выгрузки модулей ядра. Отдельное внимание уделено анализу информации о модулях и их параметрах на примере файловой системы ext4 и подсистемы Bluetooth. Кроме того, была выполнена проверка версии ядра и изучен процесс обновления ядра операционной системы. Полученные результаты позволяют лучше понять принципы модульной архитектуры ядра Linux и практику администрирования системы.