

Отчёт по лабораторной работе №10

Основы работы с модулями ядра операционной системы

Ришард Когенгар

Содержание

1	Цель работы	5
2	Ход выполнения	6
2.1	Управление модулями ядра из командной строки	6
2.2	Загрузка и выгрузка модуля ext4	9
2.3	Загрузка модулей ядра с параметрами на примере bluetooth	11
2.4	Обновление ядра системы	13
2.5	Вывод	15
3	Контрольные вопросы	16

Список иллюстраций

2.1	Просмотр устройств и связанных модулей командой <code>lspci -k</code>	7
2.2	Просмотр загруженных модулей ядра	8
2.3	Загрузка <code>ext4</code> и просмотр информации о модуле	10
2.4	Выгрузка <code>ext4</code> и ошибка при выгрузке <code>xfs</code>	11
2.5	Загрузка <code>bluetooth</code> и базовая информация о модуле	12
2.6	Параметры модуля <code>bluetooth</code> и выгрузка	13
2.7	Проверка версии ядра и попытка обновления через <code>dnf</code>	14
2.8	Проверка версии ядра после перезагрузки и сведения о системе .	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 терминале получены полномочия администратора командой `su -`, что позволило выполнять операции загрузки/выгрузки модулей ядра и просмотр служебной информации без ограничений прав доступа.
2. Для определения устройств системы и связанных с ними драйверов/модулей выполнена команда `lspci -k`.

По выводу видно, какие PCI-устройства обнаружены и какой модуль ядра обслуживает каждое из них (строки **Kernel driver in use** и **Kernel modules**).

В системе определены, в частности:

- IDE-контроллер Intel PIIX4, обслуживаемый драйвером **ata_piix** (модули `ata_piix`, `ata_generic`);
- видеоконтроллер SVGA, обслуживаемый драйвером **vmwgfx** (модуль `vmwgfx`);
- сетевой адаптер Intel PRO/1000 MT, обслуживаемый драйвером **e1000** (модуль `e1000`);
- сервисное устройство VirtualBox Guest Service, обслуживаемое драйвером **vboxguest** (модуль `vboxguest`);

- аудиоконтроллер AC'97, обслуживаемый драйвером **snd_intel8x0** (модуль snd_intel8x0);
 - SATA-контроллер AHCI, обслуживаемый драйвером **ahci** (модуль ahci).
- Таким образом, команда позволяет установить связь «устройство → активный драйвер/модуль ядра», что важно для диагностики и управления поддержкой оборудования.

```
root@rishardkogengar:/home/rishard# 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
        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 PIIX4 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@rishardkogengar:/home/rishard# █
```

Рис. 2.1: Просмотр устройств и связанных модулей командой `lspci -k`

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

В выводе отображаются: **имя модуля**, его **размер** и поле **Used by** (количество/перечень зависимостей, т.е. кто использует модуль).

В списке присутствуют как аппаратные драйверы (например, `e1000`, `vmwgfx`,

ahci, ata_piix), так и системные модули (например, dm_mod, fuse, crc32c*, i2c_piix4), что подтверждает корректную загрузку основных подсистем.

```

-----rishardkogengar:/home/rishard/-----
root@rishardkogengar:/home/rishard# lsmod | sort
ac97_bus          12288  1 snd_ac97_codec
ahci              57344  3
ata_generic      12288  0
ata_piix         45056  1
be2iscsi         204800  0
bnx2i            94208  0
cdrom            90112  2 isofs,sr_mod
cnic             90112  1 bnx2i
crc32c_intel     12288  0
crc32_pclmul     12288  0
crct10dif_pclmul 12288  1
cxgb4            684032  1 cxgb4i
cxgb4i           69632  0
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_msr
intel_rapl_msr   20480  0
intel_uncore_frequency_common 16384  0
intel_vsec       20480  1 intel_pmc_core
iscsi_boot_sysfs 20480  1 be2iscsi
iscsi_tcp        28672  0
isofs            69632  1
joydev           28672  0
libahci          69632  1 ahci

```

Рис. 2.2: Просмотр загруженных модулей ядра

2.2 Загрузка и выгрузка модуля ext4

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

До загрузки модуль в списке отсутствовал (совпадений не найдено), что указывает на то, что поддержка ext4 не была активирована как модуль в текущей сессии.

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

Повторная проверка `lsmod | grep ext4` показала появление модуля **ext4**, а также связанных с ним модулей **jbd2** и **mbcache**, которые используются ext4 как зависимости (журналирование и кэширование метаданных).

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

В выводе отражены ключевые характеристики:

- **filename** — путь к файлу модуля (`.../kernel/fs/ext4/ext4.ko.xz`);
- **license/description/author** — лицензия, описание и авторы;
- **alias** — альтернативные идентификаторы (в т.ч. совместимость с ext2/ext3 через алиасы);
- **depends** — зависимости (`jbd2, mbcache`);
- **vermagic** — строка совместимости с версией/сборкой ядра;
- **sig_*** и **signature** — сведения о цифровой подписи модуля.

Отдельно отмечено, что у модуля **нет параметров** (в `modinfo` отсутствуют строки `parm:`), то есть при загрузке `modprobe ext4` нельзя настраивать его поведение через параметры модуля.

```

root@rishardkogengar:/home/rishard#
root@rishardkogengar:/home/rishard# lsmod | grep ext4
root@rishardkogengar:/home/rishard# modprobe ext4
root@rishardkogengar:/home/rishard# lsmod | grep ext4
ext4                1187840    0
mbcache              16384     1  ext4
jbd2                 217088     1  ext4
root@rishardkogengar:/home/rishard# modinfo ext4
filename:            /lib/modules/6.12.0-124.27.1.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:            6.12.0-124.27.1.el10_1.x86_64 SMP preempt mod_unload modversions
sig_id:              PKCS#7
signer:              Rocky Linux kernel signing key
sig_key:             02:F7:96:08:65:A1:E8:61:62:5C:11:B3:91:CA:BA:58:40:2F:66:03
sig_hashalgo:        sha256
signature:           72:A4:FD:F2:20:7F:91:93:CB:B6:00:1D:3F:5C:32:75:F8:F6:16:67:
                    FF:0B:BB:0C:03:CE:0B:BB:37:1A:F3:A0:30:A3:00:CB:F7:00:F7:A7:

```

Рис. 2.3: Загрузка ext4 и просмотр информации о модуле

7. Выполнена попытка выгрузки модуля ext4 командой `modprobe -r ext4`.

Команда была введена несколько раз; после этого проверка `lsmod | grep ext4` не вывела совпадений, что означает успешную выгрузку модуля (при отсутствии активных ext4-разделов/использований, удерживающих модуль в памяти).

8. Выполнена попытка выгрузки модуля xfs командой `modprobe -r xfs`.

Система вернула сообщение об ошибке: **modprobe: FATAL: Module xfs is in use.**, что означает невозможность выгрузки из-за активного использования (как правило, когда корневая/рабочая файловая система или смонтированные разделы используют XFS).

```

root@rishardkogengar:/home/rishard#
root@rishardkogengar:/home/rishard# modprobe -r ext4
root@rishardkogengar:/home/rishard# modprobe -r ext4
root@rishardkogengar:/home/rishard# lsmod | grep ext4
root@rishardkogengar:/home/rishard# modprobe -r xfs
modprobe: FATAL: Module xfs is in use.
root@rishardkogengar:/home/rishard# modprobe -r xfs
modprobe: FATAL: Module xfs is in use.
root@rishardkogengar:/home/rishard# █

```

Рис. 2.4: Выгрузка ext4 и ошибка при выгрузке xfs

2.3 Загрузка модулей ядра с параметрами на примере bluetooth

9. Проверено наличие модуля Bluetooth командой `lsmod | grep bluetooth`.
До загрузки совпадений не было, что указывает на отсутствие загруженного модуля `bluetooth` в текущий момент.
10. Выполнена загрузка модуля командой `modprobe bluetooth`.
После этого команда `lsmod | grep bluetooth` показала появление модуля **bluetooth**, а также зависимого модуля **rftkill**, используемого подсистемой для программной блокировки/разблокировки радиointерфейсов.
11. Выполнена команда `modinfo bluetooth` для просмотра сведений о модуле.
В выводе отображены: путь к файлу модуля (`.../kernel/net/bluetooth/bluetooth.ko.xz`), описание и версия подсистемы, зависимость **depends: rftkill**, а также цифровая подпись и строка совместимости `vermagic`.

```

root@rishardkogengar:/home/rishard#
root@rishardkogengar:/home/rishard# lsmod | grep bluetooth
root@rishardkogengar:/home/rishard# modprobe bluetooth
root@rishardkogengar:/home/rishard# lsmod | grep bluetooth
bluetooth                1118208  0
rfkill                    40960  4 bluetooth
root@rishardkogengar:/home/rishard# modinfo bluetooth
filename:                 /lib/modules/6.12.0-124.27.1.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:                 6.12.0-124.27.1.el10_1.x86_64 SMP preempt mod_unload modversions
sig_id:                   PKCS#7
signer:                   Rocky Linux kernel signing key
sig_key:                  02:F7:96:08:65:A1:E8:61:62:5C:11:B3:91:CA:BA:58:40:2F:66:03
sig_hashalgo:             sha256
signature:                5D:6E:37:24:52:47:31:5F:85:BC:5C:1D:5D:CF:F6:78:F9:26:3A:73:
                          F5:0C:9E:B4:11:4C:9C:0D:DD:A5:94:68:A9:95:6F:D9:24:D8:B4:85:
                          BE:71:B4:D4:10:06:CF:30:A3:2B:8D:D7:13:69:77:5C:2D:EF:6B:33:
                          F4:09:AC:CF:E1:51:56:01:86:7D:86:82:CE:40:E9:A0:CB:29:CC:59:
                          37:C3:4F:87:BC:84:29:6E:52:DA:49:3D:ED:E5:85:37:F2:9E:BD:09:

```

Рис. 2.5: Загрузка bluetooth и базовая информация о модуле

12. В `modinfo bluetooth` дополнительно представлены параметры модуля (строки `parm:`), которые могут задаваться при загрузке:

- `disable_esco` — отключение создания eSCO-соединений (логический параметр `bool`);
- `disable_ertm` — отключение режима Enhanced Retransmission Mode (логический параметр `bool`);
- `enable_ecred` — включение режима Enhanced Credit Flow Control (логический параметр `bool`).

Данные параметры могут применяться при загрузке через `modprobe bluetooth <param>=0/1`, позволяя управлять отдельными возможностями стека Bluetooth.

13. Выполнена выгрузка модуля командой `modprobe -r bluetooth`.

Ошибок не возникло, что означает успешное удаление модуля из памяти при отсутствии активных зависимостей/использований.

```
37:65:2F:BF:6B:72:DD:25:6F:91:67:AD:CC:5F:71:E9:54:A7:11:5F:
E0:74:E4:E1:ED:BD:1B:13:21:91:B2:42:4D:4D:BE:D0:B3:56:37:A5:
DB:A0:8C:33:73:D0:32:5A:1D:DC:8F:D3:09:8E:A8:F2:42:70:84:CE:
1F:8F:4C:6F:AD:96:EE:DF:F8:5B:22:83:01:FB:1A:99:4D:A3:E9:6B:
31:1F:8C:4F:E8:65:EA:82:AC:6F:7C:D4:2C:1E:C5:01:84:4A:79:67:
14:88:74:12:75:3B:7B:66:5E:16:B6:0D:53:EB:17:24:0F:EC:E0:13:
C6:97:B0:8F
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@rishardkogengar:/home/rishard# modprobe -r bluetooth
root@rishardkogengar:/home/rishard# modprobe -r bluetooth
root@rishardkogengar:/home/rishard# █
```

Рис. 2.6: Параметры модуля bluetooth и выгрузка

2.4 Обновление ядра системы

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

Используется ядро **6.12.0-124.27.1.el10_1.x86_64**.

15. Выведен список пакетов, относящихся к ядру, командой `dnf list kernel`.

В системе установлены пакеты ядра, включая версию **6.12.0-124.27.1.el10_1**, а также присутствует более ранний пакет (из окружения установки), что типично для систем, где сохраняются несколько версий ядра.

16. Выполнено обновление метаданных и пакетов командой `dnf upgrade --refresh`, после чего выполнено обновление ядра `dnf update kernel`.

По выводу **Dependencies resolved / Nothing to do / Complete!** система актуальна, обновления для ядра на момент проверки отсутствуют.

```

root@rishardkogengar:/home/rishard# uname -r
6.12.0-124.27.1.el10_1.x86_64
root@rishardkogengar:/home/rishard# dnf list kernel
Rocky Linux 10 - BaseOS                                12 kB/s | 4.3 kB    00:00
Rocky Linux 10 - AppStream                              16 kB/s | 4.3 kB    00:00
Rocky Linux 10 - Extras                                9.5 kB/s | 3.1 kB    00:00
Installed Packages
kernel.x86_64                                           6.12.0-124.8.1.el10_1      @anaconda
kernel.x86_64                                           6.12.0-124.27.1.el10_1      @baseos
root@rishardkogengar:/home/rishard# dnf upgrade --refresh
Rocky Linux 10 - BaseOS                                4.1 kB/s | 4.3 kB    00:01
Rocky Linux 10 - AppStream                              16 kB/s | 4.3 kB    00:00
Rocky Linux 10 - Extras                                12 kB/s | 3.1 kB    00:00
Dependencies resolved.
Nothing to do.
Complete!
root@rishardkogengar:/home/rishard# dnf update kernel
Last metadata expiration check: 0:00:14 ago on Wed 21 Jan 2026 11:34:43 AM MSK.
Dependencies resolved.
Nothing to do.
Complete!

```

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

17. После перезагрузки выполнена повторная проверка версии ядра `uname -r` и вывод сведений о системе `hostnamectl`.

Версия ядра осталась **6.12.0-124.27.1.el10_1.x86_64**, что соответствует состоянию «обновлений нет».

Команда `hostnamectl` также подтверждает: ОС **Rocky Linux 10.1 (Red Quartz)**, виртуализация **oracle/VirtualBox**, архитектура **x86-64**, а статическое имя хоста — **rishardkogengar.localdomain**.

```

root@rishardkogengar:/home/rishard#
root@rishardkogengar:/home/rishard# uname -r
6.12.0-124.27.1.el10_1.x86_64
root@rishardkogengar:/home/rishard# hostnamectl
  Static hostname: rishardkogengar.localdomain
    Icon name: computer-vm
    Chassis: vm 🖥️
  Machine ID: d90867af7659402b989cb6f4b1bcab43
    Boot ID: 45effe82dfb64048b8c4bea95dc76a6b
  Product UUID: e8645d61-c521-3f41-bf2b-b5d1198e8109
  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
    Kernel: Linux 6.12.0-124.27.1.el10_1.x86_64
    Architecture: x86-64
  Hardware Vendor: innotek GmbH
  Hardware Model: VirtualBox
  Hardware Serial: VirtualBox-615d64e8-21c5-413f-bf2b-b5d1198e8109
  Firmware Version: VirtualBox
    Firmware Date: Fri 2006-12-01
    Firmware Age: 19y 1month 3w
root@rishardkogengar:/home/rishard# █

```

Рис. 2.8: Проверка версии ядра после перезагрузки и сведения о системе

2.5 Вывод

В ходе работы были изучены основные приёмы управления модулями ядра Linux из командной строки. Получены навыки определения аппаратных устройств и соответствующих им драйверов, просмотра загруженных модулей, а также их загрузки и выгрузки с учётом зависимостей. Установлено, что модули, активно используемые системой (например, `xfst`), не могут быть выгружены. Рассмотрена загрузка модулей с параметрами на примере подсистемы Bluetooth. Также выполнена проверка и обновление ядра операционной системы Rocky Linux, подтверждена актуальность установленной версии ядра и корректность работы системы после перезагрузки.

3 Контрольные вопросы

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

Для просмотра текущей версии загруженного и используемого ядра применяется команда
`uname -r`.

Она выводит точную версию ядра Linux, с которым в данный момент работает операционная система.

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

Более подробную информацию можно получить с помощью следующих команд:
`uname -a` — выводит расширенные сведения, включая версию ядра, архитектуру и дату сборки.

`hostnamectl` — отображает информацию об ОС, версии ядра, архитектуре и типе виртуализации.

Дополнительно сведения о пакетах ядра можно посмотреть командой `dnf list kernel`.

3. Какая команда показывает список загруженных модулей ядра?

Для просмотра всех загруженных в память модулей ядра используется команда `lsmod`.

Она отображает имя модуля, его размер и список зависимостей (или количество использующих его модулей).

4. Какая команда позволяет вам определять параметры модуля ядра?

Для определения параметров модуля ядра применяется команда

```
modinfo <имя_модуля>.
```

В выводе параметры отображаются строками `parm:` с указанием имени параметра и его назначения.

5. Как выгрузить модуль ядра?

Выгрузка модуля ядра выполняется командой

```
modprobe -r <имя_модуля>
```

или альтернативно `rmmod <имя_модуля>`.

При этом модуль будет выгружен только в том случае, если он не используется системой.

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

Если система сообщает, что модуль используется, необходимо:

- определить, какие процессы или подсистемы используют данный модуль;
- остановить службы или размонтировать файловые системы, связанные с модулем;
- после этого повторить попытку выгрузки;
- если модуль критически важен (например, файловая система корня), его выгрузка невозможна без перезагрузки системы.

7. Как определить, какие параметры модуля ядра поддерживаются?

Поддерживаемые параметры модуля определяются командой

```
modinfo <имя_модуля>.
```

Все доступные параметры перечисляются в выводе в виде строк `parm:`, где указано имя параметра, его назначение и тип.

8. Как установить новую версию ядра?

Установка или обновление ядра выполняется средствами пакетного менеджера:

```
dnf update kernel — обновление пакета ядра до доступной версии;
```

```
dnf upgrade --refresh — обновление системы с обновлением метаданных репозитория.
```

После установки нового ядра требуется перезагрузка системы и выбор нового ядра при старте ОС.