

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №10**

дисциплина: Основы администрирования операционных систем

Студент: Накова Амина Михайловна

Студ. билет № 1132232887

Группа: НПИбд-02-23

**МОСКВА**

2025 г.

## Цель работы:

Целью данной работы является получение навыков работы с утилитами управления модулями ядра операционной системы.

## Выполнение работы:

### Управление модулями ядра из командной строки:

Запустим терминал и получим полномочия администратора: **su -**. Посмотрим, какие устройства имеются в нашей системе и какие модули ядра с ними связаны:

**lspci -k** (Рис. 1.1):

```
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: piix4_smbus
    Kernel modules: i2c_piix4
00:0d.0 SATA controller: Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E) SATA Controller
```

**Рис. 1.1.** Запуск терминала и получение полномочий администратора, просмотр имеющихся устройств в нашей системе и модулей ядра, связанных с ними.

Теперь посмотрим, какие модули ядра загружены: **lsmod | sort** (Рис. 1.2):

```

ac97_bus      16384 1 snd_ac97_codec
ahci          49152 2
ata_generic   16384 0
ata_piix      45056 1
cdrom         81920 2 isofs,sr_mod
crc32c_intel  24576 1
crc32_pclmul  16384 0
crct10dif_pclmul 16384 1
dm_log        20480 2 dm_region_hash,dm_mirror
dm_mirror     28672 0
dm_mod        184320 9 dm_log,dm_mirror
dm_region_hash 24576 1 dm_mirror
drm           622592 6 vmwgfx,drm_kms_helper,drm_ttm_helper,ttm
drm_kms_helper 200704 1 vmwgfx
drm_ttm_helper 16384 1 vmwgfx
e1000         167936 0
fb_sys_fops   16384 1 drm_kms_helper
fuse          176128 3
ghash_clmulni_intel 16384 0
i2c_piix4     28672 0
intel_pmc_core 53248 0
intel_pmc_core_pltdrv 16384 0
intel_powerclamp 20480 0
intel_rapl_common 28672 1 intel_rapl_ms
intel_rapl_ms 20480 0

```

**Рис. 1.2.** Просмотр загруженных модулей ядра.

Посмотрим, загружен ли модуль `ext4`: **`lsmod | grep ext4`** (модуль не загружен). Затем загрузим модуль ядра `ext4`: **`modprobe ext4`** и убедимся, что модуль загружен, посмотрев список загруженных модулей: **`lsmod | grep ext4`**. Посмотрим информацию о модуле ядра `ext4`: **`modinfo ext4`** и обратим внимание, что у этого модуля нет параметров (Рис. 1.3):

```

ext4                962560 0
mbcache             16384 1 ext4
jbd2                 167936 1 ext4
[root@ismakhorin ~]# modinfo ext4
filename:           /lib/modules/5.14.0-162.6.1.el9_1.0.1.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:        9.1
srcversion:          53809B61F040B3F04B7A060
depends:             mbcache,jbd2
retpoline:          Y
intree:             Y
name:               ext4

```

**Рис. 1.3.** Просмотр информации о наличии загруженного модуля ext4, загрузка модуля и проверка просмотром списка загруженных модулей, просмотр информации о модуле ядра ext4.

На следующем шаге попробуем выгрузить модуль ядра ext4: **modprobe -r ext4**. Команду потребовалось ввести два раз. Далее попробуем выгрузить модуль ядра xfs: **modprobe -r xfs**. Обратите внимание, что мы получаем сообщение об ошибке, поскольку модуль ядра в данный момент используется (Рис. 1.4):

```
modprobe: FATAL: Module xfs is in use.
```

**Рис. 1.4.** Попытка выгрузки модуля ядра ext4 и модуля ядра xfs.

### Загрузка модулей ядра с параметрами:

Запустим терминал и получим полномочия администратора. Посмотрим, загружен ли модуль bluetooth: **lsmod | grep bluetooth** (модуль Bluetooth не установлен). Загрузим модуль ядра bluetooth: **modprobe bluetooth** и посмотрим список модулей ядра, отвечающих за работу с Bluetooth: **lsmod | grep bluetooth**. Посмотрим информацию о модуле bluetooth: **modinfo bluetooth** (Рис. 2.1).

```

bluetooth      856064  0
rfkill         36864  4 bluetooth
[root@ismakhorin ~]# modinfo bluetooth
filename:      /lib/modules/5.14.0-162.6.1.el9_1.0.1.x86_64/kernel/net/bluetooth/bluet
ooth.ko.xz
alias:         net-pf-31
license:       GPL
version:       2.22
description:   Bluetooth Core ver 2.22
author:        Marcel Holtmann <marcel@holtmann.org>
rhelversion:   9.1
srcversion:    E73B067C8E2B06674D73D58
depends:        rfkill
retpoline:     Y
intree:        Y
name:          bluetooth
vermagic:      5.14.0-162.6.1.el9_1.0.1.x86_64 SMP preempt mod_unload modversions
sig_id:        PKCS#7
signer:        CentOS Stream kernel signing key
sig_key:       63:1A:A4:97:93:59:E0:50:D1:91:64:62:F3:FD:42:15:5F:F3:4C:0E
sig_hashalgo:  sha256
signature:     A6:58:AA:D0:CA:92:7D:24:15:61:BC:F9:14:15:BC:39:D0:CC:32:5B:
                2A:DD:F7:6B:FA:C9:C9:CE:1F:C8:49:23:19:C4:43:15:FD:36:18:F1:
                EA:88:87:DB:3B:9C:34:D4:89:DF:D9:69:50:56:82:A7:EA:12:15:3F:
                FB:40:F1:5B:A2:8E:76:07:A2:B5:F6:CD:F8:F0:F4:75:49:54:EF:02:
                8C:7D:90:E0:88:73:22:0F:E8:F5:5F:02:EA:B9:8E:45:4A:8B:94:04:

```

**Рис. 2.1.** Запуск терминала и получение полномочий администратора, просмотр информации о наличии загруженного модуля bluetooth, загрузка модуля ядра bluetooth и просмотр списка модулей ядра, отвечающих за работу с bluetooth.

Просмотр информации о модуле bluetooth.

Выгрузим модуль ядра bluetooth: **modprobe -r bluetooth** (Рис. 2.2).

```

]# modprobe -r bluetooth
]#

```

**Рис. 2.2.** Выгрузка модуля ядра bluetooth.

## Обновление ядра системы:

Запустим терминал и получим полномочия администратора: **su —**. Посмотрим версию ядра, используемую в операционной системе: **uname -r**. Выведем на

экран список пакетов, относящихся к ядру операционной системы: **dnf list kernel**  
(Рис. 3.1):

```
Installed Packages
kernel.x86_64           5.14.0-70.17.1.el9_0      @anaconda
kernel.x86_64           5.14.0-70.30.1.el9_0      @baseos
kernel.x86_64           5.14.0-162.6.1.el9_1.0.1  @baseos
```

**Рис. 3.1.** Просмотр версии ядра, используемой в ОС. Вывод на экран списка пакетов, относящихся к ядру ОС.

Обновим систему, чтобы убедиться, что все существующие пакеты обновлены, так как это важно при установке/обновлении ядер Linux и избежания конфликтов: **dnf upgrade --refresh**

Обновим ядро операционной системы, а затем саму операционную систему (Рис. 3.3):

**dnf update kernel**

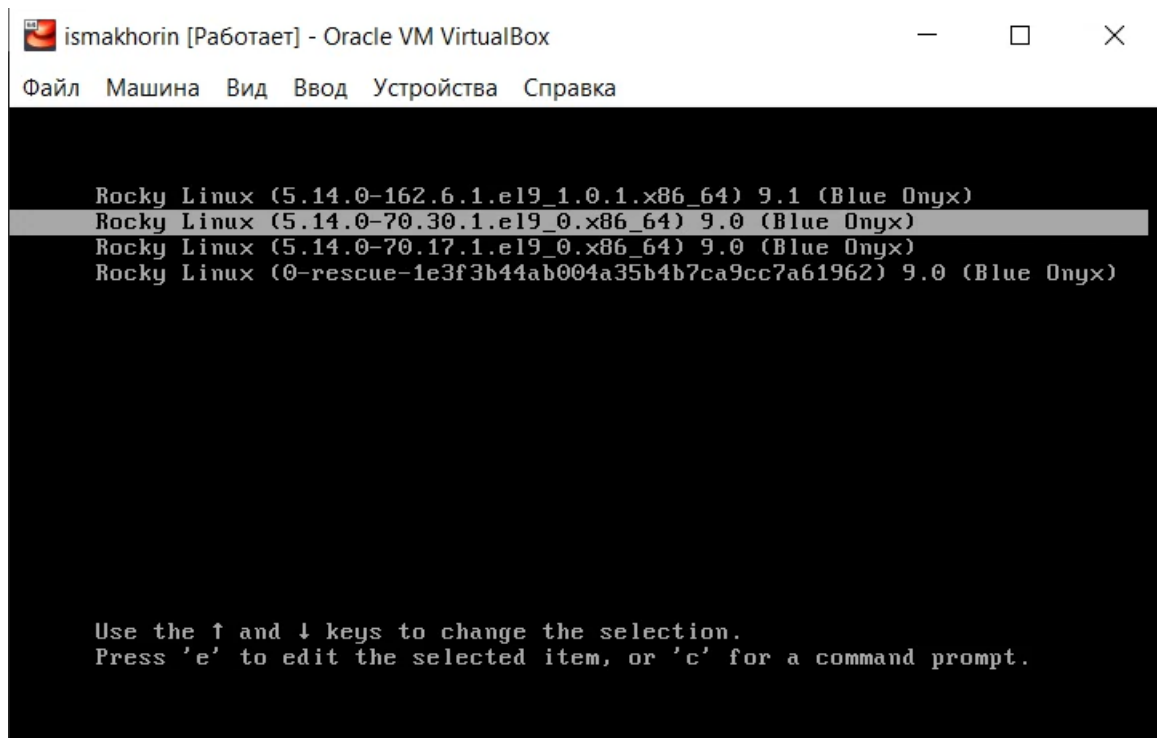
**dnf update**

**dnf upgrade --refresh**

Затем перегрузим систему и при загрузке выберем новое ядро (Рис. 3.4).

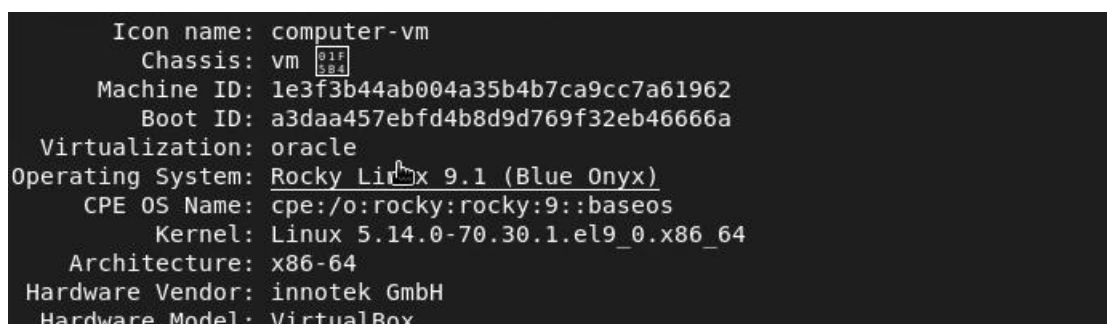
```
Rocky Linux 9 - BaseOS        6.2 kB/s | 3.6 kB    00:00
Rocky Linux 9 - AppStream      7.7 kB/s | 4.1 kB    00:00
Rocky Linux 9 - Extras        5.7 kB/s | 2.9 kB    00:00
Dependencies resolved.
Nothing to do.
Complete!
```

**Рис. 3.3.** Обновление ядра ОС, а затем самой ОС. Запуск перезагрузки системы.



**Рис. 3.4.** Выбор нового ядра.

Запустим терминал и получим полномочия администратора: **su -**. После чего посмотрим версию ядра, используемую в операционной системе: **uname -r hostnamectl** (Рис. 3.5):



**Рис. 3.5.** Запуск терминала и получение полномочий администратора, просмотр версии ядра, используемой в ОС.

## Ответы на контрольные вопросы:

1. Какая команда показывает текущую версию ядра, которая используется на вашей системе? **uname -r**

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

3. Какая команда показывает список загруженных модулей ядра? **lsmod | sort**

4. Какая команда позволяет вам определять параметры модуля ядра?  
**modprobe <имя модуля> <параметры> = <значение модуля>**

5. Как выгрузить модуль ядра? **modprobe -r <модуль>**

6. Что вы можете сделать, если получите сообщение об ошибке при попытке выгрузить модуль ядра? **Сперва выгружаем тот модуль, который занимает нужный нам модуль, а потом выгружаем первоначальный.**

7. Как определить, какие параметры модуля ядра поддерживаются? **modinfo <модуль>**

8. Как установить новую версию ядра?

**1) Обновим систему, чтобы убедиться, что все существующие пакеты обновлены, так как это важно при установке/обновлении ядер Linux и избежания конфликтов:**



**dnf upgrade --refresh**

**2) Обновим ядро операционной системы, а затем саму операционную систему:**

**dnf update kernel**

**dnf update dnf upgrade --refresh**

**3) Перегрузим систему. При загрузке выберем новое ядро**

**Вывод:**

В ходе выполнения лабораторной работы были получены навыки работы с утилитами управления модулями ядра операционной системы.