

# **Лабораторная работа №10**

**Отчет**

Славинский Владислав Вадимович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выводы</b>	<b>13</b>
<b>4</b>	<b>Ответы на контрольные вопросы</b>	<b>14</b>

## Список иллюстраций

2.1	Получение root прав . . . . .	6
2.2	Устройства в системе . . . . .	7
2.3	Загруженные ядра . . . . .	7
2.4	Проверка модуля ext4 . . . . .	7
2.5	Загрузка модуля ext4 . . . . .	8
2.6	Просмотр информации о модуле ядра ext4 . . . . .	8
2.7	Выгрузка ext4 . . . . .	9
2.8	Выгрузка xfs . . . . .	9
2.9	Проверка модуля bluetooth . . . . .	9
2.10	Загрузка bluetooth . . . . .	9
2.11	Информация о модуле bluetooth . . . . .	10
2.12	Выгрузка модуля bluetooth . . . . .	10
2.13	Версия ядра Linux . . . . .	10
2.14	Вывод списка пакетов . . . . .	10
2.15	Обновление системы . . . . .	11
2.16	Обновление ядра ОС, ОС . . . . .	11
2.17	Выбор нового ядра . . . . .	12

## **Список таблиц**

# 1 Цель работы

Получить навыки работы с утилитами управления модулями ядра операционной системы.

## 2 Выполнение лабораторной работы

Запустим терминал и получим полномочия администратора: su - (рис. 2.1)

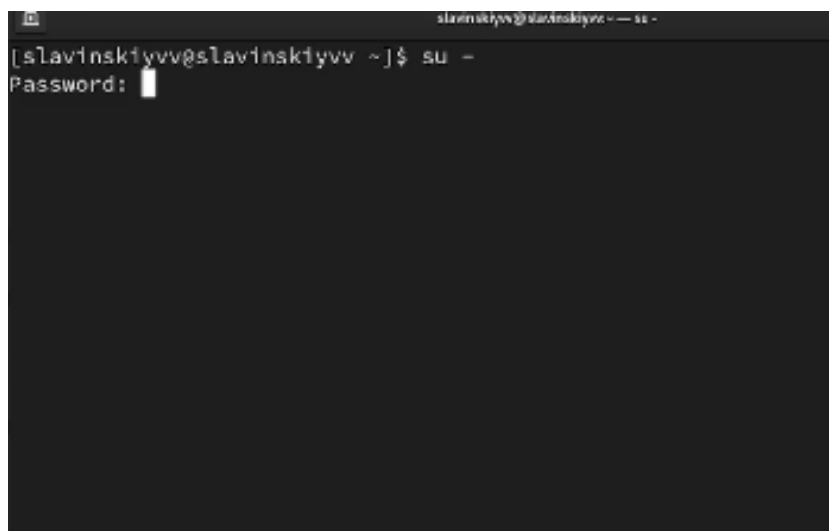


Рис. 2.1: Получение root прав

Посмотрим, какие устройства имеются в нашей системе и какие модули ядра с ними связаны: lspci -k. Эта команда выводит нам список устройств, которые подключены через шину PCI, а также модули ядра, которые используются для работы этих устройств. Например VGA compatible controller - это виртуальный видеоконтроллер VMware. А драйвер vmgfx - это драйвер графического адаптера VMware для гостевых систем. (рис. 2.2)

```

[slavinskiyvv@slavinskiyvv ~]$ su -
Password:
[root@slavinskiyvv ~]# 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

```

Рис. 2.2: Устройства в системе

Посмотрим, какие модули ядра загружены: `lsmod | sort`. (рис. 2.3)

```

    Kernel driver in use: ahci
    Kernel modules: ahci
[root@slavinskiyvv ~]# lsmod | sort
ac97_bus                12288  1 snd_ac97_codec
ahci                     49152  3
ata_generic             16384  0
ata_piix                 45056  1
cdrom                    90112  2 iso9660, sr_mod
crc32c_intel             24576  1
crc32_pclmul             12288  0
crt10dif_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  7 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_pmc_core            122880  0

```

Рис. 2.3: Загруженные ядра

Посмотрим, загружен ли модуль `ext4`: `lsmod | grep ext4`. Модуль не загружен, поэтому нам нужно его загрузить. (рис. 2.4)

```

vmwgfx                  475136  2
wmi                     45056  1 video
xfs                     2686976  2
[root@slavinskiyvv ~]# lsmod | grep ext4
[root@slavinskiyvv ~]#

```

Рис. 2.4: Проверка модуля `ext4`

Загрузим модуль ядра ext4: modprobe ext4. Проверяем, загрузился ли он. (рис. 2.5)

```
xtfs                2686976    2
[root@slavinskiyvv ~]# lsmod | grep ext4
[root@slavinskiyvv ~]# modprobe ext4
[root@slavinskiyvv ~]# lsmod | grep ext4
ext4                 1191936    0
mbcache              16384    1 ext4
jbd2                  221184    1 ext4
[root@slavinskiyvv ~]#
```

Рис. 2.5: Загрузка модуля ext4

Далее посмотрим информацию о модуле ядра ext4: modinfo ext4. Команда выводит полную информацию о модуле ядра Linux. Модуль ядра ext4 обеспечивает поддержку одноименной файловой системы в Linux. (рис. 2.6)

```
author:      Remy Card, Stephen Tweedie, Andrew Morton, Andreas Dilger, Theodore T'so and others
alias:       fs-ext4
alias:       ext3
alias:       fs-ext3
alias:       ext2
alias:       fs-ext2
rhelversion: 9.6
srcversion:  30C9EEDC227E3D26EE09D87
depends:      jbd2,mbcache
retpoline:   Y
intree:      Y
name:        ext4
vermagic:    5.14.0-570.39.1.el9_6.x86_64 SMP preempt mod_unload modversions
sig_id:      PKCS#7
signer:      Rocky kernel signing key
sig_key:     63:AD:10:02:33:B5:DE:70:E4:AB:58:C1:A4:DB:58:4B:31:65:42:9E
sig_hashalgo: sha256
signature:   63:37:DC:96:17:12:13:4E:8F:DB:64:D2:9A:69:53:4D:05:32:C1:CB:
             D8:1D:55:B3:38:09:75:79:76:D2:CD:09:13:B7:FE:A2:1A:A0:28:8E:
             0E:FC:17:41:A8:48:B5:79:28:E2:F1:9D:FC:72:6A:C3:1F:F1:F9:25:
             5E:27:A7:96:43:C8:B9:54:52:87:7D:9C:9E:28:BF:0F:02:CA:C7:87:
             18:09:DC:4B:81:01:8F:D3:81:B2:31:82:09:E4:B8:FD:D2:36:E3:03:
             BA:3E:A2:5D:A4:9B:4B:55:20:FC:80:22:8F:3E:1A:AC:CD:97:91:CF:
```

Рис. 2.6: Просмотр информации о модуле ядра ext4

Выгрузим модуль ядра ext4: modprobe -r ext4. В первый раз нам не удалось выгрузить, потому что ext4 использует другой модуль ядра, а именно crc32c\_intel. Во второй раз получилось, потому что модуль crc20c\_intel больше не использовался напрямую системой, либо часть функционала ext4 была временно неактивна. (рис. 2.7)

```

D8:94:84:9C
[root@slavinskiyvv ~]# modprobe -r ext4
modprobe: FATAL: Module crc32c_intel is in use.
[root@slavinskiyvv ~]# modprobe -r ext4
[root@slavinskiyvv ~]#

```

Рис. 2.7: Выгрузка ext4

Потом попробуем выгрузить модуль ядра xfs: `modprobe -r xfs`. Здесь же мы не можем уже выгрузить модуль ядра xfs. (рис. 2.8)

```

[root@slavinskiyvv ~]# modprobe -r ext4
[root@slavinskiyvv ~]# modprobe -r xfs
modprobe: FATAL: Module xfs is in use.
[root@slavinskiyvv ~]#

```

Рис. 2.8: Выгрузка xfs

Дальше посмотрим, загружен ли модуль bluetooth: `lsmod | grep bluetooth`. Видим, что у нас не загружен. (рис. 2.9)

```

modprobe: FATAL: Module xfs is in use.
[root@slavinskiyvv ~]# lsmod | grep bluetooth
[root@slavinskiyvv ~]#

```

Рис. 2.9: Проверка модуля bluetooth

Загрузим модуль bluetooth: `modprobe bluetooth`. И дальше проверим список модулей ядра: `lsmod | grep bluetooth` (рис. 2.10)

```

[root@slavinskiyvv ~]# lsmod | grep bluetooth
[root@slavinskiyvv ~]# modprobe bluetooth
[root@slavinskiyvv ~]# lsmod | grep bluetooth
bluetooth          1114112  0
rfkill              40960  4 bluetooth
[root@slavinskiyvv ~]#

```

Рис. 2.10: Загрузка bluetooth

Посмотрим информацию о модуле bluetooth: `modinfo bluetooth`. Команда вывела информацию о модуле ядра bluetooth. У нас есть такие параметры как: `disable eSCO support`, `disable retransmission mode`, `enable enhanced credit flow control mode`.

Первый параметр - это отключение типа аудиоканала bluetooth. Второй параметр - это отключение расширенного режима повторной передачи, и третий параметр включает улучшенный режим управления потоком данных. (рис. 2.11)

```

90:83:EA:12:30:40:EC:M1:1E:0C:0A:33:70:7A:70:3A:71:00:00:7F:
70:78:12:FF:73:17:64:DA:F5:F1:68:EB:EB:E1:29:F8:78:C3:7F:C8:
33:B1:65:E3:85:2A:FF:1B:27:4F:C8:4F:62:39:D6:22:C9:FA:6F:34:
DE:87:5D:47:FF:D3:86:C7:F6:2D:A3:8F:EF:33:0C:10:5E:29:8F:88:
DD:53:EB:97:B8:1F:77:74:DF:1E:A6:FA:86:27:01:FC:56:CB:D9:E5:
C7:6C:C8:DC:B3:0F:E9:6E:61:6C:5C:F8:F7:C8:C8:2D:E7:AB:05:A1:
F5:C0:BB:CB:35:CE:23:11:B8:25:4C:2E:E8:A3:AB:DA:B2:D4:B0:E7:
63:F5:2D:AC:10:42:1C:86:AB:15:E7:51:4E:0E:11:3A:E0:CD:9C:9C:
43:16:8D:F3:BC:AE:3A:61:F6:D2:15:50:3C:C0:13:66:68:01:32:90:
5F:CA:DF:E9
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@slavinskiyvv ~]#

```

Рис. 2.11: Информация о модуле bluetooth

Выгрузим модуль bluetooth. (рис. 2.12)

```

[root@slavinskiyvv ~]# modprobe -r bluetooth
[root@slavinskiyvv ~]#

```

Рис. 2.12: Выгрузка модуля bluetooth

Дальше посмотрим версию ядра Linux, используемую в операционной системе: `uname -r`. (рис. 2.13)

```

[root@slavinskiyvv ~]# modprobe -r bluetooth
[root@slavinskiyvv ~]# uname -r
5.14.0-570.39.1.el9_6.x86_64
[root@slavinskiyvv ~]#

```

Рис. 2.13: Версия ядра Linux

Выведем на экран список пакетов, относящихся к ядру операционной системы: `dnf list kernel`. (рис. 2.14)

```

[root@slavinskiyvv ~]# dnf list kernel
Last metadata expiration check: 3:05:28 ago on Fri 07 Nov 2025 08:39:12 PM MSK.
Installed Packages
kernel.x86_64                5.14.0-570.17.1.el9_6      @anaconda
kernel.x86_64                5.14.0-570.37.1.el9_6      @baseos
kernel.x86_64                5.14.0-570.39.1.el9_6      @baseos
Available Packages
kernel.x86_64                5.14.0-570.56.1.el9_6      baseos
[root@slavinskiyvv ~]#

```

Рис. 2.14: Вывод списка пакетов

Обновим систему, чтобы убедиться, что все существующие пакеты обновлены, чтобы избежать конфликтов: `dnf upgrade --refresh`. (рис. 2.15)

```
kernel.x86_64                               5.14.0-570.58.1.el9_6          baseos
[root@slavinskiyvv ~]# dnf upgrade --refresh
Extra Packages for Enterprise Linux 9 - x86_64 [=====] --- B/s | 0 B
```

Рис. 2.15: Обновление системы

Обновим ядро операционной системы, а затем саму операционную систему: `dnf update kernel, dnf update, dnf upgrade --refresh`. (рис. 2.16)

```
Complete!
[root@slavinskiyvv ~]# dnf update kernel
Last metadata expiration check: 0:11:11 ago on Fri 07 Nov 2025 11:45:19 PM MSK.
Dependencies resolved.
Nothing to do.
Complete!
[root@slavinskiyvv ~]# dnf update
Last metadata expiration check: 0:11:10 ago on Fri 07 Nov 2025 11:45:19 PM MSK.
Dependencies resolved.
Nothing to do.
Complete!
[root@slavinskiyvv ~]# dnf upgrade --refresh
Extra Packages for Enterprise Linux 9 - x86_64          48 kB/s | 38 kB    00:00
Extra Packages for Enterprise Linux 9 openh264 (From Cisco) - x86_64 6.3 kB/s | 993 B    00:00
Rocky Linux 9 - BaseOS          17 kB/s | 4.1 kB    00:00
Rocky Linux 9 - AppStream       13 kB/s | 4.5 kB    00:00
Rocky Linux 9 - Extras          14 kB/s | 2.9 kB    00:00
Dependencies resolved.
Nothing to do.
Complete!
[root@slavinskiyvv ~]#
```

Рис. 2.16: Обновление ядра ОС, ОС

Перезагружаем систему, выбираем новое ядро и проверяем изменения: `uname -r, hostnamectl`. И как видим, версия ядра изменилась. (рис. 2.17)

```
[slavinskiyv@slavinskiyv ~]$ uname -r
5.14.0-570.58.1.el9_6.x86_64
[slavinskiyv@slavinskiyv ~]$ hostnamectl
  Static hostname: slavinskiyv
        Icon name: computer-vm
        Chassis: vm RT
        Machine ID: bd69652ae92748ae5b335d52b03b91df
        Boot ID: 85e6ba4786bc4f2493a82765b9ebbd15
        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
[slavinskiyv@slavinskiyv ~]$ □
```

Рис. 2.17: Выбор нового ядра

## **3 Выводы**

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

## 4 Ответы на контрольные вопросы

1. `uname -r`
2. `uname -a`
3. `lsmod`
4. `modinfo`
5. `modprobe -r`
6. Найти, что использует модуль `lsmod | grep <`, завершить процессы, которые используют модуль, если возможно, попробовать выгрузить зависимые модули перед целевым модулем
7. Через команду `modinfo` ищем строку `parm`
8. `dnf update kernel`, `dnf update`