

Лабораторная работа № 10. Основы работы с модулями ядра операционной системы

Отчёт

Сергеев Даниил Олегович

Содержание

1 Цель работы	5
2 Задание	6
3 Ход выполнения лабораторной работы	7
3.1 Управление модулями ядра из командной строки	7
3.2 Загрузка модулей ядра с параметрами	11
3.3 Обновление ядра системы	14
4 Ответы на контрольные вопросы	16
5 Вывод	18
Список литературы	19

Список иллюстраций

3.1 Список устройств и связанных модулей ядра	7
3.2 Загруженные модули ядра kernel	9
3.3 Загрузка модуля ext4	9
3.4 Подробная информация о ext4	10
3.5 Выгрузка ext4 и xfs	11
3.6 Загрузка модуля bluetooth и список модулей ядра	12
3.7 Информация о bluetooth	13
3.8 Версия ядра и обновление системы	14
3.9 Обновление ядра и системы	15
3.10 Просмотр новой версии ядра	15

Список таблиц

1 Цель работы

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

2 Задание

- Продемонстрируйте навыки работы по управлению модулями ядра.
- Продемонстрируйте навыки работы по загрузке модулей ядра с параметрами.

3 Ход выполнения лабораторной работы

3.1 Управление модулями ядра из командной строки

Запустим терминал и зайдем в учетную запись администратора (su -). Посмотрим, какие устройства имеются в системе на данный момент и какие модули ядра с ними связаны.

```
lspci -k
```

```
[dosergeev@dosergeev ~]$ su -
Password:
[root@dosergeev ~]# 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
```

Рис. 3.1: Список устройств и связанных модулей ядра

Рассмотрим полученную информацию.

Общая форма записи устройства:

- Шаблон: адрес класс: производитель [вендор:устройство] название [доп_инфо] (ревизия)
- Пример: 00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
 1. 00:01.0 - Номер шины PCI: 00; Номер устройства: 01; Номер функции: 0;
 2. ISA bridge - мост для использования шин ISA;
 3. Intel Corporation - производитель Intel;
 4. 82371SB PIIX3 ISA - имя устройства;
 5. [Natoma/Triton II] - устройство входит в состав чипсета Natoma/Triton II;

Для модулей ядра, используемых устройством:

- Kernel driver in use - драйвер ядра, который используется в данный момент;
- Kernel modules - список всех модулей, которых использует данное устройство;

Теперь посмотрим, какие модули ядра загружены, для этого воспользуемся командой:

```
lsmod | sort
```

```
[root@dosergeev ~]# lsmod | sort
ac97_bus           12288  1 snd_ac97_codec
ahci              49152  3
ata_generic       16384  0
ata_piix          45056  1
cdrom             90112  2 isoofs,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 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
```

Рис. 3.2: Загруженные модули ядра kernel

Команда вывела список всех модулей, содержащий также размер модулей в памяти, количество используемых экземпляров модуля и зависимости от других модулей.

Проверим, загружен ли модуль ext4 и если нет, загрузим его и убедимся, что он загрузится:

```
lsmod | grep ext4
modprobe ext4
lsmod | grep ext4
```

```
[root@dosergeev ~]# lsmod | grep ext4
[root@dosergeev ~]# modprobe ext4
[root@dosergeev ~]# lsmod | grep ext4
ext4                1191936  0
mbcache            16384  1 ext4
jbd2              221184  1 ext4
```

Рис. 3.3: Загрузка модуля ext4

Модуль ext4 успешно загружен. Проверим информацию о нем:

```
modinfo ext4
```

```
[root@dosergeev ~]# modinfo ext4
filename:      /lib/modules/5.14.0-570.39.1.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:    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:5B: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:FF:A2:1A:A0:28:BE:
             0E:FC:17:41:A8:48:B5:79:28:E2:F1:9D:FC:72:6A:C3:1F:F1:F9:25:
             5E:27:A7:98:43:C8:B9:54:52:B7:7D:9C:9E:28:BF:0F:02:CA:C7:B7:
             10:09:DC:4B:81:01:8E:D3:81:B2:31:82:09:E4:B8:FD:D2:38:E3:03:
             BA:3E:A2:5D:A4:9B:4B:55:20:FC:80:22:8F:3E:1A:AC:CD:97:91:CF:
             32:D1:47:8E:A7:9F:80:22:F8:7C:74:25:7A:CC:54:8D:C1:56:44:37:
             F7:3B:6E:98:0C:49:83:A4:77:B8:E8:FD:38:17:DB:95:99:EF:14:55:
             20:6D:25:C5:88:F7:AA:CC:88:87:BD:D9:0D:64:61:84:B9:30:4C:65:
             DE:E0:22:DF:C9:ED:F8:85:19:78:43:4D:6E:30:0A:D7:B8:54:6A:5C:
             95:D2:A6:FD:76:4D:63:23:0A:FC:22:66:98:73:92:8C:E6:2A:8B:81:
             C2:CD:5E:8B:5F:38:8D:3D:9B:74:28:C0:32:52:1F:3B:4F:6F:C6:AC:
             BE:9F:BE:6A:FE:D6:A2:67:22:FC:DC:5F:90:52:F0:2C:2A:5A:40:
             91:66:CD:1D:2E:32:57:FC:54:9B:87:78:3E:A6:11:F0:B9:F6:AF:9F:
             81:97:B7:91:7F:97:90:08:E3:2D:3C:37:1D:4F:B6:59:65:BD:0C:5E:
             CF:4D:B5:DE:DB:2D:5E:B2:ED:8F:2D:7E:C7:C4:63:53:76:BB:6A:5B:
             F9:8B:45:78:31:64:DA:BA:65:FF:01:06:F0:DA:6D:D4:63:8F:4D:7C:
             18:31:21:C2:51:7B:3A:EB:90:B2:63:67:12:2F:1A:DB:7F:E2:D9:90:
             B0:59:2E:B2:A4:53:3B:2A:F2:B3:B1:A3:F6:FE:D2:91:2A:A5:C1:E6:
             D8:94:84:9C
```

Рис. 3.4: Подробная информация о ext4

Опишем выведенную командой информацию.

- filename - полный путь к сжатому файлу модуля ядра в формате .xz;
- softdep - мягкие зависимости;
- license - тип лицензии;
- description - краткое описание модуля;
- author - разработчик(и) модуля;
- alias - псевдонимы модуля (альтернативные имена);
- rheVersion - версия Red Hat Enterprise Linux, для которой собран модуль;
- srcversion - уникальный хеш исходного кода модуля;
- depends - жесткие зависимости;
- retpoline - преключатель о наличии защиты от уязвимости Spectre;
- intree - переключатель о том, является ли модуль частью основного дерева

ядра Linux;

- vermagic - совместь версий ядра;
- sig_id - тип системы цифровой подписи;
- signer - владелец ключа подписи;
- sig_key - отпечаток ключа подписи;
- sig_hashalgo - алгоритм хеширования для подписи;
- signature - полная цифровая подпись модуля в hex-формате;

После загрузки модуля, попробуем его выгрузить:

```
modprobe -r ext4
```

Для отключения модуля потребовалось ввести команду второй раз.

Теперь попробуем выгрузить модуль ядра xfs:

```
modprobe -r xfs
```

Мы получили сообщение об ошибке, так как модуль ядра в данный момент используется.

```
[root@dosergeev ~]# modprobe -r ext4
modprobe: FATAL: Module crc32c_intel is in use.
[root@dosergeev ~]# modprobe -r ext4
[root@dosergeev ~]# lsmod | grep ext4
[root@dosergeev ~]# modprobe -r xfs
modprobe: FATAL: Module xfs is in use.
[root@dosergeev ~]# lsmod | grep xfs
xfs              2686976  2
libcrc32c          12288  4 nf_conntrack,nf_nat,nf_tables,xfs
[root@dosergeev ~]# modprobe -r xfs
modprobe: FATAL: Module xfs is in use.
```

Рис. 3.5: Выгрузка ext4 и xfs

3.2 Загрузка модулей ядра с параметрами

Посмотрим, загружен ли модуль bluetooth в системе, если нет, то загрузим его:

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

После загрузки посмотрим список модулей ядра, отвечающих за работу с Bluetooth:

```
lsmod | grep bluetooth
```

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

Рис. 3.6: Загрузка модуля bluetooth и список модулей ядра

Теперь посмотрим информацию о модуле bluetooth:

```
modinfo bluetooth
```

```
[root@dosergeev ~]# modinfo bluetooth
filename:      /lib/modules/5.14.0-570.39.1.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:    C6ED64B96F8C9A31FC91B17
depends:       rfkill
retpoline:     Y
intree:        Y
name:          bluetooth
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:5B:4B:31:65:42:9E
sig_hashalgo: sha256
signature:    B2:BD:44:95:4D:D2:F6:C3:46:2C:8F:E8:87:24:88:F9:4A:04:93:91:
              5F:6D:A2:29:C9:8E:2C:22:3C:79:8E:2F:A2:9A:17:65:30:B7:0A:42:
              4F:C8:DB:18:A2:0D:F5:54:03:D6:82:9E:3E:43:50:81:0B:24:A8:FF:
              65:EB:E9:D1:B0:B0:B7:20:B5:83:D2:A0:71:23:1F:74:E5:3A:52:ED:
              8C:61:29:68:F4:5E:F6:56:2D:40:85:2F:8D:2B:1E:88:92:07:8B:5D:
              AC:40:D3:9C:CD:DA:ED:D7:67:59:B6:DB:99:2B:D7:89:BF:AD:DC:35:
              11:30:7B:EB:F3:95:00:16:34:EA:B4:FB:26:42:5D:0F:5F:AD:68:E4:
              13:08:5F:5A:98:3B:3E:38:00:0D:BA:C8:C1:5F:BF:F5:F4:F5:E3:E3:
              0E:55:D1:9B:E1:26:EF:CF:C5:00:17:D5:D7:9E:6E:A9:F7:54:E2:27:
              EC:FF:68:1F:91:20:D5:84:BE:B2:CF:68:DB:E2:36:C5:D1:93:F5:E3:
              9D:65:EA:F2:3B:A8:EC:41:1E:6C:6A:35:78:7A:76:9A:F1:C6:E8:47:
              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:B6: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)
```

Рис. 3.7: Информация о bluetooth

В отличие от ext4, у модуля bluetooth имеются параметры, а именно:

- disable_esco - Отключить создание соединений eSCO;
- disable_ertm - Отключить расширенный режим повторной передачи;
- enable_ecred - Включить расширенный режим управления потоком на основе кредитов;

Выгрузим модуль ядра bluetooth.

```
modprobe -r bluetooth
```

3.3 Обновление ядра системы

В том же терминале с полномочиями администратора посмотрим версию ядра, используемую в операционной системе, а также выведем на экран список пакетов, относящихся к ядру операционной системы.

```
uname -r
```

```
dnf list kernel
```

Обновим пакеты системы:

```
dnf upgrade --refresh
```

```
[root@dosergeev ~]# uname -r
5.14.0-570.39.1.el9_6.x86_64
[root@dosergeev ~]# dnf list kernel
Last metadata expiration check: 1:51:22 ago on Sat 08 Nov 2025 04:07:18 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          5.14.0-570.58.1.el9_6           baseos
[root@dosergeev ~]# dnf upgrade --refresh
Extra Packages for Enterprise Linux 9 - x86_64
Extra Packages for Enterprise Linux 9 openh264 (From Cisco) - x86_64
packages for the GitHub CLI
Rocky Linux 9 - BaseOS
Rocky Linux 9 - AppStream
Rocky Linux 9 - CRB
Rocky Linux 9 - Extras
Dependencies resolved.
=====
Package          Architecture 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
=====

```

Рис. 3.8: Версия ядра и обновление системы

Ядро имеет версию 5.14.0-570.39.1.el9_6.x86_64

Обновим ядро операционной системы, а затем саму операционную систему:

```
dnf update kernel
```

```
dnf update
```

```
dnf upgrade --refresh
```

```
[root@dosergeev ~]# dnf update kernel
Last metadata expiration check: 0:30:13 ago on Sat 08 Nov 2025 05:59:50 PM MSK.
Dependencies resolved.
Nothing to do.
Complete!
[root@dosergeev ~]# dnf update
Last metadata expiration check: 0:30:25 ago on Sat 08 Nov 2025 05:59:50 PM MSK.
Dependencies resolved.
Nothing to do.
Complete!
[root@dosergeev ~]# dnf upgrade --refresh
Extra Packages for Enterprise Linux 9 - x86_64
Extra Packages for Enterprise Linux 9 openh264 (From Cisco) - x86_64
packages for the GitHub CLI
Rocky Linux 9 - BaseOS
Rocky Linux 9 - AppStream
Rocky Linux 9 - CRB
Rocky Linux 9 - Extras
Dependencies resolved.
Nothing to do.
Complete!
[root@dosergeev ~]#
```

Рис. 3.9: Обновление ядра и системы

Все пакеты и ядро уже обновлены, поэтому перегрузим систему. В моем случае новая версия ядра при загрузке выбралась автоматически, поэтому пропустим этап выбора ядра.

Посмотрим новую версию ядра.

```
uname -r
hostnamectl
```

```
[dosergeev@dosergeev ~]$ uname -r
5.14.0-570.58.1.el9_6.x86_64
[dosergeev@dosergeev ~]$ hostnamectl
Static hostname: dosergeev.localdomain
Icon name: computer-vm
Chassis: vm [01]
Machine ID: 30fa918ba5df4b1f8dee15f25a4580d7
Boot ID: 05d297dfd1eb4715ab7a7e1eb1b2ac3c
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
[dosergeev@dosergeev ~]$
```

Рис. 3.10: Просмотр новой версии ядра

Теперь система имеет версию 5.14.0-570.58.1.el9_6.x86_64

4 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`
4. Какая команда позволяет вам определять параметры модуля ядра?
 - `modinfo <модуль>`
5. Как выгрузить модуль ядра?
 - `modprobe -r <модуль>`
 - или `rmmmod <модуль>`
6. Что вы можете сделать, если получите сообщение об ошибке при попытке выгрузить модуль ядра?
 - Проверить зависимости модуля, если они есть, то выгрузить их. После чего снова попытаться выгрузить модуль.

7. Как определить, какие параметры модуля ядра поддерживаются?

- modinfo <модуль> | grep parm

8. Как установить новую версию ядра?

- dnf upgrade –refresh - обновим систему
- dnf update kernel - обновим ядро на новую версию
- dnf update - проверим обновление ядра
- dnf upgrade –refresh - проверим обновление системы
- После выполнения данных команд необходимо перезапустить систему, чтобы выбрать новое ядро.

5 Вывод

В результате выполнения лабораторной работы я узнал как проверять, какие модули и устройства загружены в систему, научился загружать и выгружать модули с параметрами и без и понял как корректно обновлять ядро системы.

Список литературы

1. Kulyabov, Korolykova. Лабораторная работа № 10. Основы работы с модулями ядра операционной системы. https://esystem.rudn.ru/pluginfile.php/2843503/mod_resource/content/4/011-kernel.pdf; RUDN.