



РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 10

Управление модулями ядра

Студент: Эйвази Мани

Группа: НПИбд-03-24

Студенческий билет №: 1032245107

Цель работы

Получить практические навыки работы с модулями ядра операционной системы Linux. Освоить методы загрузки, выгрузки, получения информации о модулях, а также обновления самого ядра системы.

Первый шаг: Определение устройств в системе и модулей ядра, которые их обслуживают. Вывод команды показывает список всех PCI-устройств. Для каждого устройства указывается драйвер (модуль ядра) в строке Kernel driver in use: и доступные модули в строке Kernel modules: Например:

- Ethernet controller: Realtek Semiconductor Co., Ltd. ... -> Kernel driver in use: r8169, Kernel modules: r8169
- Это показывает, что для сетевой карты используется модуль r8169.

```
root@localhost:/home/manieyvazi# 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
```

Второй шаг: Получение списка всех модулей, загруженных в ядро. Выводится отсортированный список модулей. Каждая строка содержит имя модуля, его размер в памяти и список модулей, которые от него зависят. Например:

```
ext4          <размер> 2 <зависимости>
crc16         <размер> 1 ext4
```

```
Kernel modules: ahci
root@localhost:/home/manieyvazi# 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
crt10dif_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
```

Третий шаг: Поиск, загрузка и проверка состояния модуля ext4. После выполнения `modprobe ext4` модуль появляется в выводе `lsmod`. Это означает, что код, отвечающий за поддержку файловой системы ext4, загружен в ядро.

```
root@localhost:/home/manieyvazi# lsmod | grep ext4
root@localhost:/home/manieyvazi# modprobe ext4
root@localhost:/home/manieyvazi# lsmod | grep ext4
ext4          1187840 0
mbcache       16384  1 ext4
jbd2          217088  1 ext4
```

Четвертый шаг: Изучение деталей о модуле ext4. Вывод содержит метаданные модуля:

- filename: Путь к файлу модуля на диске (например, `/lib/modules/.../ext4.ko.xz`).
- license: GPL (лицензия).
- description: Fourth Extended Filesystem (описание).
- depends: Список модулей, от которых зависит ext4 (например, `mbcache`, `jbd2`).

- `retpoline`: Y (указание на поддержку мер безопасности).
- Поля `parm`: отсутствуют, что означает, что модуль не принимает параметров при загрузке.

Команда: `modinfo ext4`

Пятый шаг: Проверка возможности выгрузки модулей `ext4` и `xfs`.

- Модуль `ext4` может быть успешно выгружен (`rmmod ext4` также работает), если в данный момент не используется (нет смонтированных разделов `ext4`).
- Модуль `xfs` выгрузить не удастся. Система выдает ошибку: `modprobe: FATAL: Module xfs is in use..` Это означает, что в данный момент есть смонтированная файловая система XFS (скорее всего, корневой раздел `/`), поэтому ядро не позволяет выгрузить необходимый для её работы драйвер.

```
root@localhost:/home/manieyvazi# modprobe -r ext4
root@localhost:/home/manieyvazi# lsmod | grep ext4
root@localhost:/home/manieyvazi# modprobe -r xfs
```

Шестой шаг: Исследование модуля, поддерживающего Bluetooth, и его параметров. После загрузки `bluetooth` загружается целая цепочка зависимых модулей (например, `btusb`, `btrtl`). Команда `modinfo bluetooth` показывает, что у данного модуля есть параметры (`parm`)

```
modprobe: FATAL: Module xfs is in use.
root@localhost:/home/manieyvazi# lsmod | grep bluetooth
root@localhost:/home/manieyvazi# modprobe bluetooth
root@localhost:/home/manieyvazi# lsmod | grep bluetooth
bluetooth      1118208  0
rfkill         40960  4 bluetooth
root@localhost:/home/manieyvazi# modprobe -r bluetooth
root@localhost:/home/manieyvazi# uname -r
6.12.0-124.28.1.el10_1.x86_64
```

Седьмой шаг: Определение версии текущего ядра и проверка доступных обновлений. `uname-r` показывает текущую запущенную версию ядра (например, 4.18.0-...). `dnf list kernel` отображает установленную версию и последнюю доступную в репозиториях.

```
root@localhost:/home/manieyvazi# uname -r
6.12.0-124.28.1.el10_1.x86_64
root@localhost:/home/manieyvazi# dnf list kernel
^CExtra Packages for Enterprise Linux 10      [===
Extra Packages for Enterprise Linux 10 - x86_64      0.0 B/s | 0 B      00:51
Errors during downloading metadata for repository 'epel':
- Curl error (35): SSL connect error for https://mirrors.fedoraproject.org/metalink?repo=epel-z-10&arch=x86_64 [Recv failure: Connection reset by peer]
Error: Failed to download metadata for repo 'epel': Cannot prepare internal mirrorlist: Interrupted by signal
```

Восьмой шаг: Полное обновление системы с установкой нового ядра. Пакет `kernel` успешно обновлен до новой версии. При этом старая версия ядра также остается в системе (для возможности отката).

```
root@localhost:/home/manieyvazi# dnf upgrade --refresh
Extra Packages for Enterprise Linux 10 - x86_64      4.8 kB/s | 29 kB      00:05
Extra Packages for Enterprise Linux 10 - x86_64      3.8 MB/s | 5.6 MB      00:01
Rocky Linux 10 - BaseOS      12 kB/s | 4.3 kB      00:00
Rocky Linux 10 - AppStream    14 kB/s | 4.3 kB      00:00
Rocky Linux 10 - AppStream    3.5 MB/s | 2.1 MB      00:00
Rocky Linux 10 - Extras      6.1 kB/s | 3.1 kB      00:00
Dependencies resolved.
=====
Package                                Architecture      Version            Repository          Size
=====
Installing:
kernel                                x86_64            6.12.0-124.29.1.el10_1      baseos              1.4 M
Upgrading:
glibc                                x86_64            2.39-58.el10_1.7          baseos              2.1 M
glibc-all-langpacks                 x86_64            2.39-58.el10_1.7          baseos              18 M
glibc-common                         x86_64            2.39-58.el10_1.7          baseos              317 k
glibc-devel                         x86_64            2.39-58.el10_1.7          appstream           579 k
glibc-gconv-extra                   x86_64            2.39-58.el10_1.7          baseos              1.6 M
```

```
Complete!
root@localhost:/home/manieyvazi# dnf update kernel

Last metadata expiration check: 0:04:51 ago on Tue 03 Feb 2026 05:13:41 PM MSK.
Dependencies resolved.
Nothing to do.
Complete!
root@localhost:/home/manieyvazi#
root@localhost:/home/manieyvazi# dnf update

Last metadata expiration check: 0:04:54 ago on Tue 03 Feb 2026 05:13:41 PM MSK.
Dependencies resolved.
Nothing to do.
Complete!
root@localhost:/home/manieyvazi#
root@localhost:/home/manieyvazi# dnf upgrade --refresh
Extra Packages for Enterprise Linux 10 - x86_64      32 kB/s | 29 kB      00:00
Rocky Linux 10 - BaseOS      11 kB/s | 4.3 kB      00:00
Rocky Linux 10 - AppStream    8.9 kB/s | 4.3 kB      00:00
Rocky Linux 10 - Extras      11 kB/s | 3.1 kB      00:00
Dependencies resolved.
Nothing to do.
Complete!
root@localhost:/home/manieyvazi#
```

девяти шаг: Перезагрузка системы (reboot). В загрузчике GRUB автоматически выбрано новое ядро (обычно первое в списке).

- Команда проверки: `uname -r`

После перезагрузки команда `uname -r` показывает номер версии нового, только что установленного ядра. Команда `hostnamectl` также отображает эту информацию в разделе Kernel.

заключение

В ходе лабораторной работы были успешно освоены ключевые аспекты работы с модулями ядра Linux:

1. **Мониторинг и диагностика:** Навыки использования `lspci-k` для связи аппаратных устройств с драйверами (модулями ядра) и `lsmod` для просмотра всех загруженных модулей с их зависимостями.

2. Управление модулями:

- **Загрузка:** Команда `modprobe` является предпочтительным способом, так как она автоматически разрешает и загружает зависимости модуля.
- **Выгрузка:** Команда `modprobe -r` (или `rmmod`) позволяет выгрузить модуль, но только если он не используется системой в данный момент (устройство не активно, файловая система не смонтирована). Получение ошибки "Module is in use" — стандартное поведение, обеспечивающее стабильность системы.

3. **Получение информации:** Утилита `modinfo` предоставляет исчерпывающие метаданные о модуле: описание, лицензию, автора, параметры (`parm`). Наличие параметров показывает возможность тонкой настройки модуля при загрузке.

4. **Обновление ядра:** Отработан безопасный процесс обновления ядра в дистрибутивах на основе RHEL (Rocky Linux):

- Обновление репозитория и всех пакетов.
- Установка нового пакета `kernel`. Важно, что это не заменяет, а добавляет новую версию ядра.

Работа продемонстрировала модульность ядра Linux, где функциональность может динамически расширяться по мере необходимости. Понимание принципов работы с модулями критически важно для системного администратора при подключении нового оборудования, отладке проблем с драйверами и обновлении системного ПО.

1. Какая команда показывает текущую версию ядра, которая используется на вашей системе?
uname-r (только версия) или uname-a (вся информация).
2. Как можно посмотреть более подробную информацию о текущей версии ядра операционной системы?
hostnamectl (показывает среди прочего версию ядра) или cat /proc /version.
3. Какая команда показывает список загруженных модулей ядра?
lsmod.
4. Какая команда позволяет вам определять параметры модуля ядра ?
modinfo <имя_модуля>. Параметры перечислены в секции parm.
5. Как выгрузить модуль ядра?
modprobe-r <имя_модуля> или rmmod <имя_модуля>. modprobe-r предпочтительнее, так как пытается выгрузить и зависимости.
6. Что вы можете сделать, если получите сообщение об ошибке при попытке выгрузить модуль ядра?
Ошибка Module is in use означает, что модуль используется системой. Необходимо остановить использование: для сетевого драйвера — выключить интерфейс (ip link set <iface> down), для файловой системы — размонтировать раздел (umount /dev/sdX). После этого выгрузка станет возможна.
7. Как определить, какие параметры модуля ядра поддерживаются?
С помощью команды modinfo <имя_модуля> и просмотра строк, начинающихся с parm:.
8. Как установить новую версию ядра?
В дистрибутивах на основе RPM (RHEL, CentOS, Rocky, Fedora): dnf update kernel или установка конкретного пакета ядра. После установки необходима перезагрузка для запуска с новым ядром.