

Отчёт по лабораторной работе №10

Основы работы с модулями ядра операционной системы

Лабси Мохаммед

Содержание

1 Цель работы	5
2 Ход выполнения	6
2.1 Управление модулями ядра из командной строки	6
2.2 Загрузка модулей ядра с параметрами	10
2.3 Обновление ядра системы	11
3 Контрольные вопросы	14
4 Заключение	16

Список иллюстраций

2.1	Вывод команды lspci -k	7
2.2	Просмотр списка загруженных модулей	8
2.3	Просмотр информации о модуле ext4	9
2.4	Попытка выгрузки модулей ext4 и xfs	9
2.5	Информация о модуле bluetooth	10
2.6	Выгрузка модуля bluetooth	10
2.7	Просмотр версии и списка пакетов ядра	11
2.8	Обновление ядра и системы	12
2.9	Просмотр информации о системе после обновления	13

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. Запущен терминал и получены права администратора с помощью команды **su -**.

После ввода пароля открыт сеанс суперпользователя **root**.

2. Для просмотра аппаратных устройств и связанных с ними модулей ядра выполнена команда **lspci -k**.

В выводе отображены контроллеры и используемые драйверы – например, для сетевого адаптера используется модуль **e1000**, для видеоадаптера – **vmwgfx**.

```
mlabsi@mlabsi:~$ su
Password:
root@mlabsi:/home/mlabsi# 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@mlabsi:/home/mlabsi#
```

Рис. 2.1: Вывод команды `lspci -k`

3. Для просмотра всех загруженных модулей применена команда `lsmod | sort`.

Отображён список модулей, включая **e1000**, **vmwgfx**, **ahci**, **snd_intel8x0** и другие.

```

root@mlabsi:/home/mlabsi# lsmod | sort
ac97_bus           12288  1 snd_ac97_codec
ahci              57344  3
ata_generic       16384  0
ata_piix          45056  1
cdrom             90112  2 isofs,sr_mod
crc32c_intel     16384  0
crc32_pclmul     12288  0
crct10dif_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
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   126976  0
intel_rapl_common 53248  1 intel_rapl_msrm
intel_rapl_msrm  20480  0
intel_uncore_frequency_common 16384  0
intel_vsec        20480  1 intel_pmc_core
ip_set             69632  0
isofs              69632  1
joydev             28672  0
libahci            69632  1 ahci
libata            512000  4 ata_piix,libahci,ahci,ata_generic
loop              45056  0
Module           Size  Used by

```

Рис. 2.2: Просмотр списка загруженных модулей

4. Проверено наличие загруженного модуля **ext4** с помощью **lsmod | grep ext4**

— модуль был найден в системе.

Затем выполнена команда **modinfo ext4** для получения подробной информации о модуле.

Из вывода видно, что модуль отвечает за файловую систему *Fourth Extended Filesystem*, лицензирован под GPL и не имеет параметров для настройки.

Рис. 2.3: Просмотр информации о модуле ext4

5. Попытка выгрузить модуль **ext4** выполнена с помощью **modprobe -r ext4**, однако операция завершилась неудачно, поскольку модуль используется системой.

Аналогично, при попытке выгрузить модуль **xfs** выведено сообщение об ошибке *Module xfs is in use.*

```
vi.or.ud.c/.r7.wu.Mr.oW.or.62.39.13./D.w1.DE./o.10.ME.EU.ro.  
FA:7A:C2:3E:9E:63:B4:57:50:92:28:E8:95:2D:D3:44:C5:8B:F5:C8  
0A:BE:F5:96:B5:D8:DE:8F:91:DA:85:E5:BF:A2:7D:9B:2A:4C:AD:4E  
07:54:52:A9:F2:62:B6:D7:58:9F:2E:8D:6A:C2:F4:8E:98:32:EB:14:  
E6:BE:EE:3B:C5:98:36:95:53:D8:61:BD:41:8A:FF:4B:AC:49:70:B2:  
19:94:AF:2A  
root@mlabsi:/home/mlabsi#  
root@mlabsi:/home/mlabsi# modprobe -r ext4  
root@mlabsi:/home/mlabsi# modprobe -r ext4  
root@mlabsi:/home/mlabsi# modprobe -r xfs  
modprobe: FATAL: Module xfs is in use.  
root@mlabsi:/home/mlabsi#
```

Рис. 2.4: Попытка выгрузки модулей ext4 и xfs

2.2 Загрузка модулей ядра с параметрами

- Проверено наличие модуля **bluetooth** командой **lsmod | grep bluetooth** – модуль отсутствовал.

После загрузки с помощью **modprobe bluetooth** модуль стал активен.

- Информация о модуле **bluetooth** просмотрена через **modinfo bluetooth**.

В выводе указано, что модуль реализует стек Bluetooth версии 2.22, имеет зависимость от модуля **rfkill** и поддерживает параметры, управляющие созданием соединений (*disable_esco*, *disable_ertm*, *enable_ecred*).

```
root@mlabsi:/home/mlabsi# lsmod | grep bluetooth
root@mlabsi:/home/mlabsi# modprobe bluetooth
root@mlabsi:/home/mlabsi# lsmod | grep bluetooth
bluetooth           1114112  0
rfkill                  40960  4 bluetooth
root@mlabsi:/home/mlabsi# modinfo bluetooth
filename:   /lib/modules/6.12.0-55.12.1.el10_0.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: 10.0
srcversion: 5F9AA895ADC3BA3840211D1
depends:   rfkill
intree:    Y
name:     bluetooth
retpoline: Y
vermagic: 6.12.0-55.12.1.el10_0.x86_64 SMP preempt mod_unload modversions
sig_id:    PKCS#7
signer:    Rocky kernel signing key
sig_key:   57:BC:A1:34:94:00:57:12:99:67:28:D8:C0:8B:1C:4B:8C:8A:13:06
sig_hashalgo: sha256
signature: 2E:24:0B:EE:9E:E6:A3:58:91:2A:D3:87:25:A3:1B:46:06:89:06:BC:
4A:99:10:8D:7A:0F:1F:FF:97:59:FD:59:74:0R:7F:42:40:14:03:C9:
```

Рис. 2.5: Информация о модуле bluetooth

- Выгрузка модуля **bluetooth** выполнена командой **modprobe -r bluetooth**.

```
2B:8A:CF:8F:23:38:B4:7A:A9:92:49:86:D7:9F:08:7D:8A:92:7D:DD:
41:D6:4B:31:F6:45:31:D4:7E:6E:E1:D6:94:BA:D0:6A:12:CC:76:A2:
12:55:EF:6B:91:CB:45:F0:F5:7B:0B:DD:7D:6B:D7:E1:B0:13:51:40:
29:D2:1D:3B:DD:E2:BF:AC:4B:77:67:BA:C8:95:16:41:00:9A:8F:96:
21:DC:9A:C2:B2:20:31:5C:EC:90:EA:B0:24:C3:5D:B9:72:31:B8:98:
DB:A9:4D:FE
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@mlabsi:/home/mlabsi# modprobe -r bluetooth
root@mlabsi:/home/mlabsi#
```

Рис. 2.6: Выгрузка модуля bluetooth

2.3 Обновление ядра системы

1. Проверена текущая версия ядра командой **uname -r – 6.12.0-55.39.1.el10_0.x86_64.**

Список доступных пакетов ядра получен через **dnf list kernel**.

```
root@mlabsi:/home/mlabsi# 
root@mlabsi:/home/mlabsi# uname -r
6.12.0-55.12.1.el10_0.x86_64
root@mlabsi:/home/mlabsi# dnf list kernel
Rocky Linux 10 - BaseOS
Rocky Linux 10 - BaseOS
Rocky Linux 10 - AppStream
Rocky Linux 10 - AppStream
Rocky Linux 10 - Extras
Rocky Linux 10 - Extras
Installed Packages
kernel.x86_64
Available Packages
kernel.x86_64
root@mlabsi:/home/mlabsi#
```

Рис. 2.7: Просмотр версии и списка пакетов ядра

2. Выполнено обновление системы для синхронизации репозиториев и предотвращения конфликтов:

dnf upgrade –refresh

После этого проведено обновление ядра и всех пакетов:

dnf update kernel, dnf update, dnf upgrade –refresh.

По завершении вывод системы подтвердил, что все пакеты установлены и обновлены успешно.

```
Aug 02 00:00:00 mlabsi kernel: [    0.000000] Kernel: 5.15.0-1000.1.1.el10_0.x86_64
xorg-x11-server-Xwayland-24.1.5-4.el10_0.x86_64
xxd-2.9.1.083-5.el10_0.1.x86_64
yum-4.20.0-14.el10_0.rocky.0.1.noarch
zlib-ng-compat-2.2.3-1.el10.rocky.0.1.x86_64
zlib-ng-compat-devel-2.2.3-1.el10.rocky.0.1.x86_64
Installed:
kernel-6.12.0-55.39.1.el10_0.x86_64          kernel-core-6.12.0-55.39.1.el10_0.x86_64
kernel-devel-6.12.0-55.39.1.el10_0.x86_64      kernel-modules-6.12.0-55.39.1.el10_0.x86_64
kernel-modules-core-6.12.0-55.39.1.el10_0.x86_64 kernel-modules-extra-6.12.0-55.39.1.el10_0.x86_64
libatomic-14.2.1-7.el10.x86_64                  libdex-0.8.1-1.el10.x86_64

Complete!
root@mlabsi:/home/mlabsi# dnf update kernel
Last metadata expiration check: 0:00:57 ago on Sat 18 Oct 2025 01:24:24 PM MSK.
Dependencies resolved.
Nothing to do.
Complete!
root@mlabsi:/home/mlabsi# dnf update
Last metadata expiration check: 0:01:04 ago on Sat 18 Oct 2025 01:24:24 PM MSK.
Dependencies resolved.
Nothing to do.
Complete!
root@mlabsi:/home/mlabsi# dnf upgrade --refresh
Rocky Linux 10 - BaseOS                               5.2 kB/s | 4.3 kB   00:00
Rocky Linux 10 - AppStream                            14 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@mlabsi:/home/mlabsi#
```

Рис. 2.8: Обновление ядра и системы

3. После перезагрузки командой **reboot** необходимо выбрать новое ядро в меню загрузки и убедиться в обновлении версии через **uname -r** и **hostnamectl**.

```
mlabsi@mlabsi:~$ uname -r
6.12.0-55.39.1.el10_0.x86_64
mlabsi@mlabsi:~$ hostnamectl
    Static hostname: mlabsi.localdomain
          Icon name: computer-vm
            Chassis: vm 🖥
        Machine ID: c371d82aeddd4c358d0da59eb13ae51b
          Boot ID: 9fddb632c4d24b5f81acec460437d10e
      Virtualization: oracle
Operating System: Rocky Linux 10.0 (Red Quartz)
      CPE OS Name: cpe:/o:rocky:rocky:10::baseos
        OS Support End: Thu 2035-05-31
OS Support Remaining: 9y 7month 1w 4d
      Kernel: Linux 6.12.0-55.39.1.el10_0.x86_64
  Architecture: x86-64
  Hardware Vendor: innoteck GmbH
  Hardware Model: VirtualBox
Firmware Version: VirtualBox
  Firmware Date: Fri 2006-12-01
  Firmware Age: 18y 10month 2w 3d
mlabsi@mlabsi:~$ █
```

Рис. 2.9: Просмотр информации о системе после обновления

3 Контрольные вопросы

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

Для отображения версии ядра используется команда **uname -r**.

Она выводит номер текущей версии ядра, загруженного в систему.

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

Более детальная информация выводится командой **hostnamectl**, где указывается версия ядра, архитектура системы, а также дистрибутив и версия ОС.

3. **Какая команда показывает список загруженных модулей ядра?**

Для просмотра всех загруженных модулей используется команда **lsmod**.

Она отображает имя модуля, его размер и количество зависимостей.

4. **Какая команда позволяет вам определять параметры модуля ядра?**

Для получения информации о модуле и его параметрах используется команда **modinfo**.

Она показывает описание, автора, зависимости, параметры и версию модуля.

5. **Как выгрузить модуль ядра?**

Для выгрузки модуля ядра применяется команда **modprobe -r**.

Если модуль не используется другими процессами, он будет успешно удалён из памяти.

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

Если модуль используется системой, необходимо сначала остановить процессы,

которые его применяют, либо отмонтировать файловую систему, использующую данный модуль.

7. Как определить, какие параметры модуля ядра поддерживаются?

Для этого используется команда **modinfo**,

в выводе которой в разделе *parm* перечислены все поддерживаемые параметры и их типы.

8. Как установить новую версию ядра?

Новое ядро устанавливается с помощью менеджера пакетов **dnf**.

Сначала выполняется обновление репозиториев командой **dnf upgrade -refresh**,

затем обновление ядра – **dnf update kernel**,

после чего выполняется **dnf update** для обновления всей системы и последующая перезагрузка.

4 Заключение

В ходе работы освоены основные команды управления модулями ядра Linux, их загрузка, выгрузка, просмотр параметров и обновление ядра системы.