

Лабораторная работа №10

Основы администрирования операционных систем

Верниковская Е. А., НПИбд-01-23

9 ноября 2024

Российский университет дружбы народов, Москва, Россия

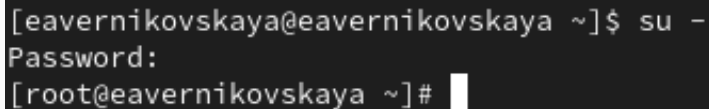
Вводная часть

Получить навыки работы с утилитами управления модулями ядра операционной системы.

1. Продемонстрировать навыки работы по управлению модулями ядра
2. Продемонстрировать навыки работы по загрузке модулей ядра с параметрами

Выполнение лабораторной работы

Запускаем терминала и получаем полномочия суперпользователя, используя *su* - (рис. 1)

A terminal window with a dark background and light gray text. The prompt is [eavernikovskaya@eavernikovskaya ~]\$ and the command entered is su -. The next line shows Password: followed by a white rectangle representing a redacted password. The final line shows the root prompt [root@eavernikovskaya ~]# followed by a white rectangle representing a redacted command.

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

Рис. 1: Режим суперпользователя

Управление модулями ядра из командной строки

Посмотрим, какие устройства имеются в нашей системе и какие модули ядра с ними связаны: *lspci -k* (рис. 2)

```
[root@eavernikovskaya ~]# 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@eavernikovskaya ~]#
```

Рис. 2: Устройства в системе и модули ядра

Управление модулями ядра из командной строки

Посмотрим, какие модули ядра загружены: `lsmod | sort` (рис. 3)

```
[root@eavernikovskaya ~]# lsmod | sort
ac97_bus          16384  1 snd_ac97_codec
ahci              49152  3
ata_generic      16384  0
ata_piix         45056  1
cdrom            90112  2 isofs,sr_mod
crc32c_intel     24576  1
crc32_pclmul     16384  0
crc32_pclmul     16384  1
dm_log           28672  2 dm_region_hash,dm_mirror
dm_mirror        32768  0
dm_mod           237568  9 dm_log,dm_mirror
dm_region_hash   28672  1 dm_mirror
drm              741376  7 vmwgfx,drm_kms_helper,drm_ttm_helper,ttm
drm_kms_helper   245760  3 vmwgfx
drm_ttm_helper   16384  1 vmwgfx
e1000            196608  0
fb_sys_fops      16384  1 drm_kms_helper
fuse             212992  3
ghash_clmulni_intel 16384  0
i2c_piix4        32768  0
intel_pmc_core   118784  0
```

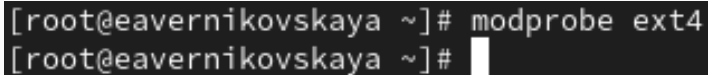
Рис. 3: Загруженные модули ядра

Посмотрим, загружен ли модуль `ext4`: `lsmod | grep ext4`. Он не загружен (рис. 4)

```
[root@eavernikovskaya ~]# lsmod | grep ext4  
[root@eavernikovskaya ~]#
```

Рис. 4: Просмотр информации о наличии загруженного модуля `ext4`

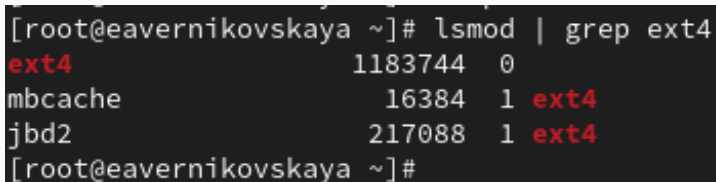
Загрузим модуль ядра `ext4` с помощью *`modprobe ext4`* (рис. 5)

A terminal window with a dark background and light gray text. The prompt is [root@eavernikovskaya ~]#. The command modprobe ext4 has been entered and executed, as indicated by the second prompt line [root@eavernikovskaya ~]# followed by a white cursor block.

```
[root@eavernikovskaya ~]# modprobe ext4
[root@eavernikovskaya ~]#
```

Рис. 5: Загрузка модуля ядра `ext4`

Проверим, что модуль загружен, посмотрев список загруженных модулей:
lsmod | grep ext4 (рис. 6)



```
[root@eavernikovskaya ~]# lsmod | grep ext4
ext4                1183744    0
mbcache              16384     1 ext4
jbd2                 217088     1 ext4
[root@eavernikovskaya ~]#
```

Рис. 6: Информация о наличии модуля ядра ext4

Посмотрим информацию о модуле ядра ext4: *modinfo ext4* (рис. 7)

```
[root@eavernikovskaya ~]# modinfo ext4
filename:       /lib/modules/5.14.0-427.13.1.el9_4.x86_64/kernel/fs/ext4/ext4.ko.xz
softdep:       prefs 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:   9.4
srcversion:    2B896FAB53D489F1C7683E6
depends:        mbcache,jbd2
retpoline:     Y
intree:        Y
name:          ext4
vermagic:      5.14.0-427.13.1.el9_4.x86_64 SMP preempt mod_unload modversions
sig_id:        PKCS#7
signer:        Rocky kernel signing key
sig_key:       40:58:2E:3B:AE:2E:C1:1E:1C:8B:04:11:83:6A:66:EF:D3:5F:8C:63
sig_hashalgo:  sha256
signature:     78:89:A1:A0:26:3B:8B:C1:F9:E6:97:16:D7:F2:DC:A1:C0:EA:8C:2C:
67:8B:03:48:32:7C:F9:1F:03:3D:ED:28:21:68:EC:A7:4A:C8:53:36:
DF:78:68:3A:12:D0:CF:99:5C:1A:3B:16:2E:2C:E8:16:25:6C:61:C4:
7D:57:84:2A:62:83:54:2A:55:82:87:03:8F:05:5B:08:BB:D1:EE:1F:
78:BD:73:35:96:97:A8:4B:DF:CE:0B:01:CE:83:41:94:15:92:A9:7E:
C8:28:63:A7:21:6F:44:48:F5:1D:2F:FE:17:16:C4:9A:F2:DA:23:F2:
31:84:FB:DC:FD:CA:25:FA:FB:F9:F2:33:53:61:C4:E0:40:BE:0F:A8:
B5:94:F1:01:89:8B:C9:5F:3A:C4:45:28:C3:83:68:00:72:A8:19:2A:
CD:E0:B0:C3:80:80:3F:A0:EF:1E:13:D5:3D:21:41:0C:C8:50:61:19:
DF:34:40:D4:E5:D2:3C:46:15:29:9F:8A:D4:57:68:63:57:C8:D1:28:
72:83:72:AF:AB:2D:33:88:CE:92:4C:59:42:8C:4E:A8:29:1D:07:C2:
81:26:0F:AF:7C:E2:AA:5F:B6:FA:D5:0A:2A:1D:B1:8F:FB:18:A0:ED:
18:EE:D6:54:33:1A:0B:D7:A4:37:23:D6:A7:A9:0C:D3:EF:42:E4:F2:
CA:EB:C3:84:39:AC:FE:F5:4B:8D:B0:E9:DC:A8:91:C4:3B:08:EF:19:
DC:3F:DC:6A:6B:93:23:B0:16:4C:67:59:FC:71:09:59:15:26:27:C2:
C3:EA:4C:48:60:ED:C3:D8:15:68:D4:91:5B:6B:26:01:51:E4:F2:53:
31:47:E8:3D:3F:8A:94:1A:E6:E4:4B:41:A1:C6:55:1C:26:C5:8C:C6:
BA:63:C8:DB:93:2D:3E:04:D9:B8:24:26:82:26:EF:36:60:A8:DE:84:
2D:35:5F:2B:45:BB:FF:20:A0:78:19:97:1C:C2:59:DF:21:E3:FD:51:
BB:FF:48:6F
[root@eavernikovskaya ~]#
```

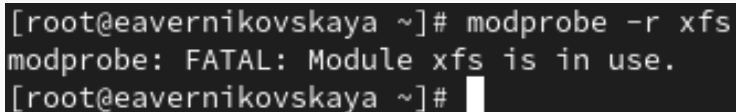
Рис. 7: Информация о модуле ядра ext4

Попробуем выгрузить модуль ядра `ext4`: `modprobe -r ext4`. Команду потребовалось ввести 2 раза. Ошибка, которая возникла при выполнении команды `modprobe -r ext4`, связана с тем, что модуль `crc32c_intel`, используемый внутри модуля `ext4`, был занят. Это значит, что какие-то процессы использовали этот модуль, и попытка его удалить вызвала ошибку (рис. 8)

```
[root@eavernikovskaya ~]# modprobe -r ext4
modprobe: FATAL: Module crc32c_intel is in use.
[root@eavernikovskaya ~]# modprobe -r ext4
[root@eavernikovskaya ~]#
```

Рис. 8: Попытка выгрузки модуля ядра `ext4`

Попробуем выгрузить модуль ядра `xfs`: `modprobe -r xfs`. Мы получили сообщение об ошибке, поскольку модуль ядра в данный момент используется (рис. 9)

A terminal window with a dark background and light gray text. The prompt is [root@eavernikovskaya ~]#. The command modprobe -r xfs is entered. The output is modprobe: FATAL: Module xfs is in use. The prompt [root@eavernikovskaya ~]# is shown again with a white cursor block.

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

Рис. 9: Попытка выгрузки модуля ядра `xfs`

Загрузка модулей ядра с параметрами

Посмотрим, загружен ли модуль bluetooth: `lsmod | grep bluetooth`. Он не загружен (рис. 10)



```
[root@eavernikovskaya ~]# lsmod | grep bluetooth
[root@eavernikovskaya ~]#
```

Рис. 10: Просмотр информации о наличии загруженного модуля bluetooth

Загрузим модуль ядра bluetooth с помощью *modprobe bluetooth* (рис. 11)

```
[root@eavernikovskaya ~]# modprobe bluetooth  
[root@eavernikovskaya ~]#
```

Рис. 11: Загрузка модуля ядра bluetooth

Загрузка модулей ядра с параметрами

Посмотрим список модулей ядра, отвечающих за работу с Bluetooth: *lsmod | grep bluetooth* (рис. 12)

```
[root@eavernikovskaya ~]# lsmod | grep bluetooth
bluetooth          1101824  0
rfkill              40960    4 bluetooth
[root@eavernikovskaya ~]#
```

Рис. 12: Список модулей ядра, отвечающих за работу с Bluetooth

Посмотрим информацию о модуле bluetooth: *modinfo bluetooth* (рис. 13)

```
[root@eavernikovskaya ~]# modinfo bluetooth
Filename: /lib/modules/5.14.0-427.13.1.el9_4.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>
rhmversion: 9.4
srcversion: 49EFE5A627A8D64249C8630
depends: rfkill
retpoline: Y
intree: Y
name: bluetooth
vermagic: 5.14.0-427.13.1.el9_4.x86_64 SMP preempt mod_unload modversions
sig_id: PKCS#7
signer: Rocky kernel signing key
sig_key: 40:50:2E:3B:AE:2E:C1:E1C:88:04:11:83:6A:66:EF:D3:5F:BC:63
sig_hashalgo: sha256
signature:
0F:EF:E3:48:68:CF:D2:D8:EB:FE:56:C3:D5:3D:06:FF:02:44:54:E6:
CA:EA:08:A7:2E:54:00:E7:55:A9:2B:82:82:DE:B2:1D:E7:4A:9C:F6:
07:E4:85:9D:24:FF:41:AF:B8:21:5A:70:E4:04:9B:03:35:F2:FC:CD:
E2:1E:8F:08:09:70:56:9D:07:8A:59:56:27:81:71:4B:54:96:FD:19:
8A:B5:8B:29:DC:91:8C:8C:B2:8F:81:C8:F1:A8:8E:E7:3C:F8:B3:86:
EA:C6:86:F7:1F:FD:AD:B8:49:64:A1:FC:1C:8C:58:E3:56:C9:12:6A:
83:0A:AE:AD:B8:67:23:DD:DA:1A:7F:D9:FE:91:29:92:69:80:1D:A6:
76:FD:2C:54:08:C5:E4:6D:A9:55:96:52:93:98:4E:3E:1E:8B:43:A8:
53:91:68:92:3C:83:34:43:3A:17:BB:BB:AC:F5:61:97:A2:2B:43:30:
EC:E2:52:18:52:86:6D:A8:98:3B:48:67:03:72:4B:E6:36:C8:8A:12:
CB:80:37:69:8A:61:D7:4F:78:AB:63:F2:4C:95:72:D2:81:D0:10:82:
78:26:91:B9:88:7C:14:C3:A5:1A:A3:A9:0A:D2:DA:F6:6B:24:5C:D8:
9C:09:01:9D:0B:CB:D5:4F:03:42:8C:C9:E2:C1:04:10:EA:29:F6:1B:
39:11:A6:10:79:11:65:8B:12:E2:FE:FD:73:8D:6B:33:0E:ED:E6:C8:59:
39:E2:8A:F9:72:10:E7:AB:2B:21:E5:50:5C:6A:67:20:9A:E6:FD:6A:
D3:3C:83:4E:81:25:35:58:3E:95:C1:1F:80:3E:A9:16:7A:3A:B1:51:
26:93:69:1B:89:6F:59:78:D2:39:87:2F:44:D3:AD:FD:9D:F8:AB:E7:
F2:53:1D:FC:DA:74:E4:FC:44:C8:3E:58:E3:68:51:5C:83:41:99:25:
31:B0:E2:77:47:5A:0E:4C:38:D0:55:A7:CE:FD:E8:48:5A:57:8A:D2:
BE:B3:BF:A6
parm: disable_esco:Disable eSCO connection creation (bool)
parm: disable_ertm:Disable enhanced retransmission mode (bool)
parm: enable_ecrc:Enable enhanced credit flow control mode (bool)
[root@eavernikovskaya ~]#
```

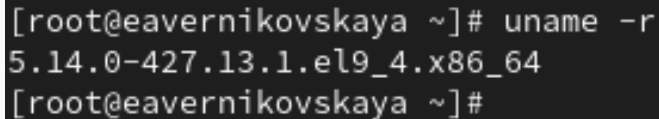
Рис. 13: Информация о модуле bluetooth

Выгрузим модуль ядра bluetooth: *modprobe -r bluetooth* (рис. 14)

```
[root@eavernikovskaya ~]# modprobe -r bluetooth  
[root@eavernikovskaya ~]#
```

Рис. 14: Выгрузка модуля ядра bluetooth

Посмотрим версию ядра, используемую в операционной системе: *uname -r* (рис. 15)



```
[root@eavernikovskaya ~]# uname -r
5.14.0-427.13.1.el9_4.x86_64
[root@eavernikovskaya ~]#
```

Рис. 15: Версия ядра, используемая в операционной системе

Выведем на экран список пакетов, относящихся к ядру операционной системы: *dnf list kernel* (рис. 16)

```
[root@eavernikovskaya ~]# dnf list kernel
Rocky Linux 9 - BaseOS                2.4 kB/s | 4.1 kB    00:01
Rocky Linux 9 - BaseOS                178 kB/s | 2.3 MB    00:13
Rocky Linux 9 - AppStream              8.2 kB/s | 4.5 kB    00:00
Rocky Linux 9 - AppStream              609 kB/s | 8.0 MB    00:13
Rocky Linux 9 - Extras                  3.5 kB/s | 2.9 kB    00:00
Installed Packages
kernel.x86_64                          5.14.0-427.13.1.el9_4 @anaconda
Available Packages
kernel.x86_64                          5.14.0-427.42.1.el9_4 baseos
[root@eavernikovskaya ~]#
```

Рис. 16: Список пакетов, относящихся к ядру операционной системы

Обновление ядра системы

Обновим систему, чтобы убедиться, что все существующие пакеты обновлены, так как это важно при установке/обновлении ядер Linux и избежания конфликтов: `dnf upgrade --refresh` (рис. 17)

```
[root@eavernikovskaya ~]# dnf upgrade --refresh
Rocky Linux 9 - BaseOS                231 B/s | 4.1 kB    00:18
Rocky Linux 9 - AppStream              5.9 kB/s | 4.5 kB    00:00
Rocky Linux 9 - Extras                 4.4 kB/s | 2.9 kB    00:00
Dependencies resolved.
=====
Package                                Architecture Version                                Repository                                Size
=====
Installing:
kernel                                x86_64     5.14.0-427.42.1.el9_4                  baseos                                    4.3 M
Upgrading:
NetworkManager                       x86_64     1:1.46.0-19.el9_4                      baseos                                    2.3 M
NetworkManager-adsl                  x86_64     1:1.46.0-19.el9_4                      baseos                                    35 k
NetworkManager-bluetooth             x86_64     1:1.46.0-19.el9_4                      baseos                                    61 k
NetworkManager-config-server         noarch     1:1.46.0-19.el9_4                      baseos                                    20 k
NetworkManager-libnm                 x86_64     1:1.46.0-19.el9_4                      baseos                                    1.8 M
```

Рис. 17: Обновление системы

Далее обновим ядро операционной системы, а затем саму операционную систему: *dnf update kernel*, *dnf update*, и *dnf upgrade –refresh* (рис. 18), (рис. 19), (рис. 20)

```
[root@eavernikovskaya ~]# dnf update kernel
Last metadata expiration check: 0:11:42 ago on Sat 09 Nov 2024 12:05:01 AM MSK.
Dependencies resolved.
Nothing to do.
Complete!
[root@eavernikovskaya ~]# █
```

Рис. 18: Обновление ядра операционной системы

```
[root@eavernikovskaya ~]# dnf update
Last metadata expiration check: 0:12:00 ago on Sat 09 Nov 2024 12:05:01 AM MSK.
Dependencies resolved.
Nothing to do.
Complete!
[root@eavernikovskaya ~]# █
```

Рис. 19: Обновление всех установленных пакетов

Обновление ядра системы

```
[root@eavernikovskaya ~]# dnf upgrade --refresh
Rocky Linux 9 - BaseOS                                4.9 kB/s | 4.1 kB    00:00
Rocky Linux 9 - AppStream                             7.4 kB/s | 4.5 kB    00:00
Rocky Linux 9 - Extras                                4.4 kB/s | 2.9 kB    00:00
Dependencies resolved.
Nothing to do.
Complete!
[root@eavernikovskaya ~]#
```

Рис. 20: Обновление системы

После, перегрузим систему и при загрузке выберем новое ядро (рис. 21), (рис. 22)



```
[root@eavernikovskaya ~]# reboot
```

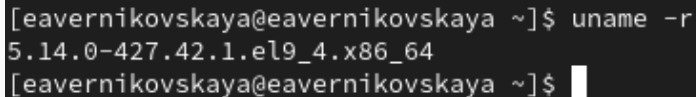
Рис. 21: Перезагрузка системы



```
[root@eavernikovskaya ~]# reboot
```

Рис. 22: Выбор нового ядра

Теперь посмотрим версию ядра, используемую в операционной системе:
uname -r и *hostnamectl* Мы видим, название ядра изменилось (рис. 23), (рис. 24)

A terminal window with a dark background. The prompt is [eavernikovskaya@eavernikovskaya ~]\$. The command 'uname -r' has been entered and executed. The output is '5.14.0-427.42.1.el9_4.x86_64'. The prompt is now [eavernikovskaya@eavernikovskaya ~]\$ followed by a white cursor block.

```
[eavernikovskaya@eavernikovskaya ~]$ uname -r
5.14.0-427.42.1.el9_4.x86_64
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 23: Новая версия ядра, используемая в операционной системе

```
[eavernikovskaya@eavernikovskaya ~]$ hostnamectl
Static hostname: eavernikovskaya.localdomain
Icon name: computer-vm
Chassis: vm 01F
5B4
Machine ID: b9202024839e4f95856600f1e460ebe9
Boot ID: 3304b38549a841209ee35cfb53fa66db
Virtualization: oracle
Operating System: Rocky Linux 9.4 (Blue Onyx)
CPE OS Name: cpe:/o:rocky:rocky:9::baseos
Kernel: Linux 5.14.0-427.42.1.el9_4.x86_64
Architecture: x86-64
Hardware Vendor: innotek GmbH
Hardware Model: VirtualBox
Firmware Version: VirtualBox
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 24: Команда hostnamectl

Подведение итогов

В ходе выполнения лабораторной работы мы получили навыки работы с утилитами управления модулями ядра операционной системы

1. Лабораторная работа №10 [Электронный ресурс] URL:
https://esystem.rudn.ru/pluginfile.php/2400729/mod_resource/content/4/011-kernel.pdf