

Лабораторная работа №1
Основы информационной безопасности

Кузьмина Мария Константиновна

Цель работы

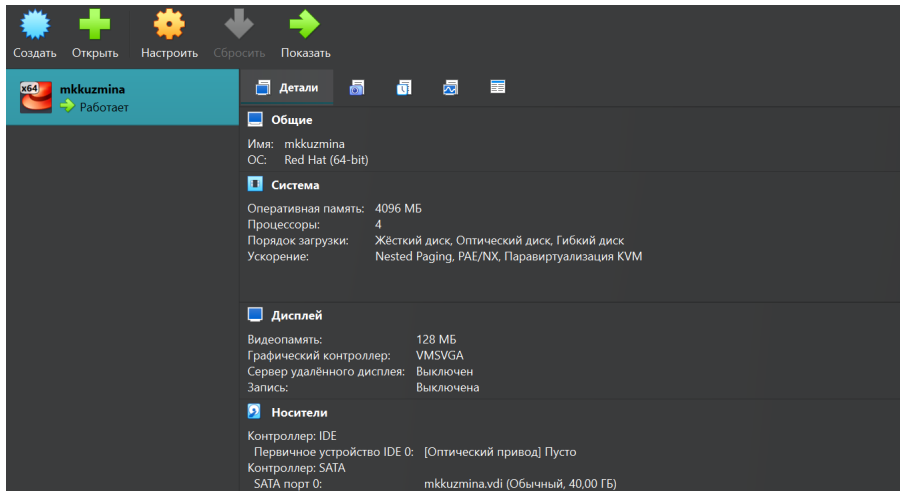
Знакомство с командами для получения информации об аппаратном обеспечении и ядре операционной системы Linux

Задание

1. Установить и настроить виртуальную машину с ОС Rocky Linux
2. Ознакомиться с выводом команды `'dmesg'`
3. Выполнить фильтрацию вывода `'dmesg'` с помощью `'grep'` для поиска информации о конкретных устройствах
4. Ответить на контрольные вопросы

Подготовка виртуального окружения

Для выполнения работы была установлена виртуальная машина с операционной системой Rocky. На скриншоте ниже представлен процесс настройки параметров виртуальной машины (имя, тип ОС, объем памяти и размер диска) (рис.1)



Использование команды dmesg

Команда используется для просмотра кольцевого буфера сообщений ядра. Она позволяет увидеть информацию об обнаруженных устройствах, драйверах и ошибках при загрузке системы

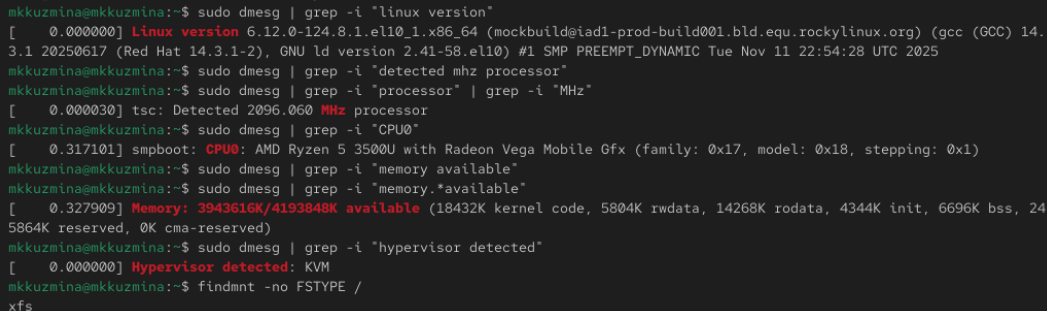
Был выполнен базовый вызов команды 'dmesg', который выводит все сообщения ядра с момента загрузки (рис.2)

```
mkkuzmina@mkkuzmina:~$ sudo dmesg
[ 0.000000] Linux version 6.12.0-124.8.1.el10_1.x86_64 (mockbuild@iad1-prod-build001.bld.equ.rockylinux.org) (gcc (GCC) 14.3.1 20250617 (Red Hat 14.3.1-2), GNU ld version 2.41-58.el10) #1 SMP PREEMPT_DYNAMIC Tue Nov 11 22:54:28 UTC 2025
[ 0.000000] Command line: BOOT_IMAGE=(hd0,gpt2)/vmlinuz-6.12.0-124.8.1.el10_1.x86_64 root=/dev/mapper/rl_10-root ro resume=UUID=6b98f970-494c-402e-9bc5-955aaca14bce rd.lvm.lv=rl_10/root rd.lvm.lv=rl_10/swap rhgb quiet
[ 0.000000] [Firmware Bug]: TSC doesn't count with P0 frequency!
[ 0.000000] BIOS-provided physical RAM map:
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable
[ 0.000000] BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000100000-0x0000000000dfffff] usable
[ 0.000000] BIOS-e820: [mem 0x000000000dfff000-0x000000000dffffff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000010000000-0x0000000011ffffff] usable
[ 0.000000] NX (Execute Disable) protection: active
[ 0.000000] APIC: Static calls initialized
[ 0.000000] SMBIOS 2.5 present.
[ 0.000000] DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
```

Фильтрация вывода dmesg

Для поиска конкретной информации, вывод команды 'dmesg' был отфильтрован с помощью утилиты 'grep'. Это позволяет, например, найти сообщения, связанные с конкретным оборудованием.

На скриншоте показан результат выполнения команды 'dmesg | grep -i "CPU0"', отображающий только строки, содержащие информацию о процессоре (рис.3)



```
mkkuzmina@mkkuzmina:~$ sudo dmesg | grep -i "linux version"
[ 0.000000] Linux version 6.12.0-124.8.1.el10_1.x86_64 (mockbuild@iad1-prod-build001.bld.equ.rockylinux.org) (gcc (GCC) 14.3.1 20250617 (Red Hat 14.3.1-2), GNU ld version 2.41-58.el10) #1 SMP PREEMPT_DYNAMIC Tue Nov 11 22:54:28 UTC 2025
mkkuzmina@mkkuzmina:~$ sudo dmesg | grep -i "detected mhz processor"
mkkuzmina@mkkuzmina:~$ sudo dmesg | grep -i "processor" | grep -i "MHz"
[ 0.000030] tsc: Detected 2096.060 MHz processor
mkkuzmina@mkkuzmina:~$ sudo dmesg | grep -i "CPU0"
[ 0.317101] smpboot: CPU0: AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx (family: 0x17, model: 0x18, stepping: 0x1)
mkkuzmina@mkkuzmina:~$ sudo dmesg | grep -i "memory available"
mkkuzmina@mkkuzmina:~$ sudo dmesg | grep -i "memory.*available"
[ 0.327909] Memory: 3943616K/4193848K available (18432K kernel code, 5804K rwd data, 14268K rodata, 4344K init, 6696K bss, 245864K reserved, 0K cma-reserved)
mkkuzmina@mkkuzmina:~$ sudo dmesg | grep -i "hypervisor detected"
[ 0.000000] Hypervisor detected: KVM
mkkuzmina@mkkuzmina:~$ findmnt -no FSTYPE /
xfs
```

Рисунок 3: снимок экрана

Выводы

В ходе выполнения лабораторной работы были изучены основные команды для диагностики системы Linux. На практике освоено использование команды 'dmesg' и фильтрация ее вывода с помощью 'grep'.

