

# Лабораторная работа №1

## Основы информационной безопасности

---

Чистов Д. М.

22 февраля 2025

Российский университет дружбы народов, Москва, Россия

Объединённый институт ядерных исследований, Дубна, Россия

## Цель работы

---

Целью данной работы является приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов

## Задания

---

1. Настройка виртуальной машины и установка операционной системы
2. Домашнее задание

## Выполнение лабораторной работы

---

Я выполнял работу дома на стационарном компьютере, пользовался VirtualBox, по заданию требуется поставить ОС Rocky, я скачал её образ с официального сайта. Перейдём к процессу настройки виртуальной машины.

# Настройка виртуальной машины и установка операционной системы

Задаю имя виртуальной машины согласно требованиям к наименованию, также задаю путь и выбираю нужный образ, скачанный с интернета.

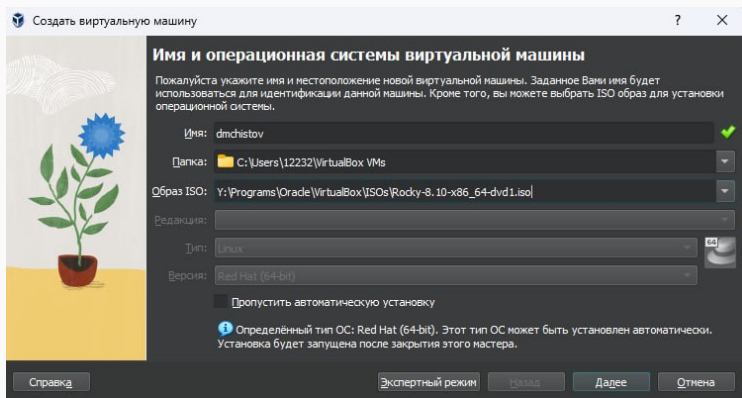


Рис. 1: Первоначальные настройки виртуальной машины



# Настройка виртуальной машины и установка операционной системы

Выбираю кол-во процессоров и выделяю нужное мне кол-во памяти.

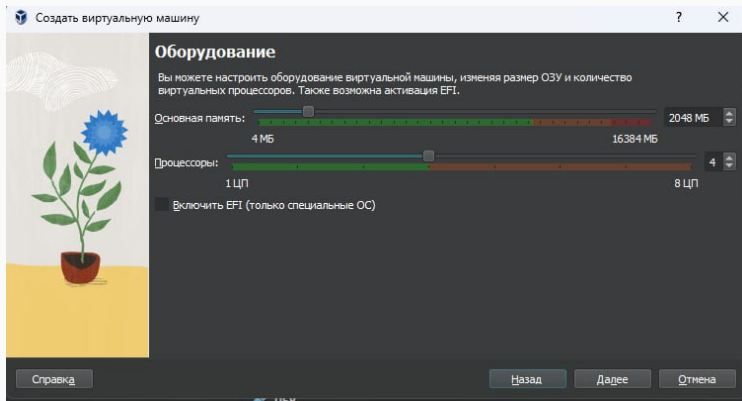


Рис. 2: Настройка виртуальной машины

# Настройка виртуальной машины и установка операционной системы

Выделяю нужное количество памяти жёсткого диска для моей виртуальной машины.

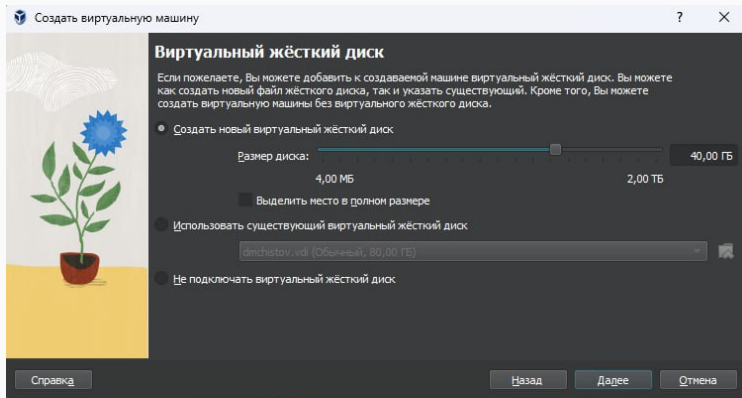
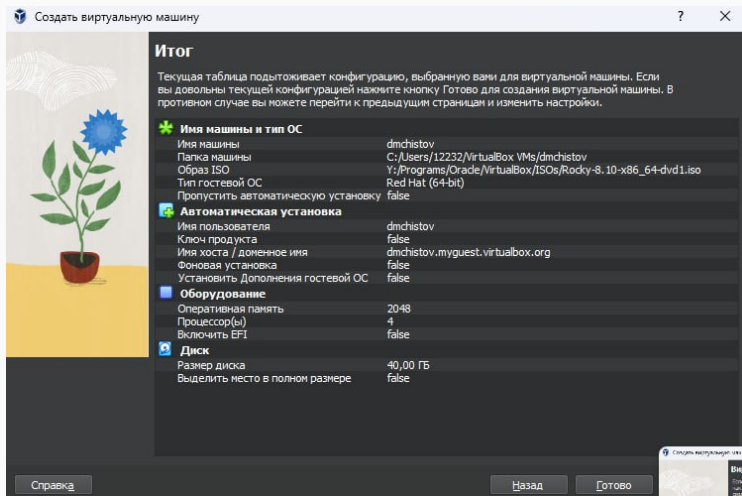


Рис. 3: Количество памяти

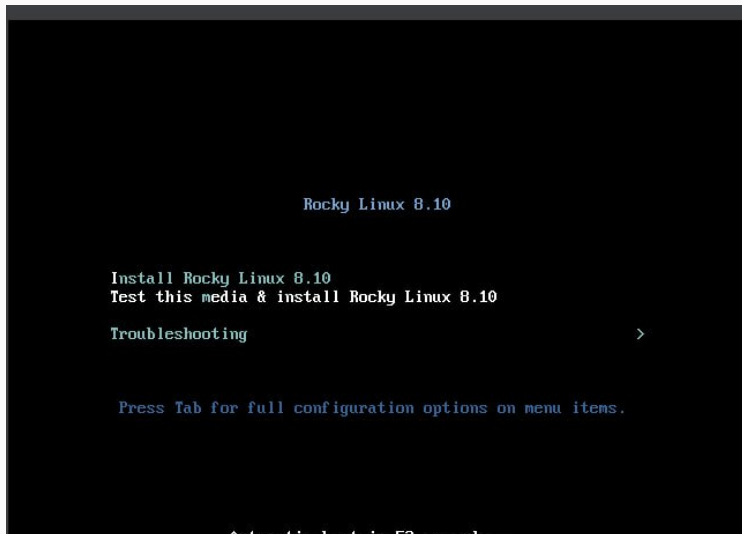
# Настройка виртуальной машины и установка операционной системы

Итоговые настройки моей виртуальной машины.



## Настройка виртуальной машины и установка операционной системы

Запускаю виртуальную машину и вижу, что установка успешно начата.



# Настройка виртуальной машины и установка операционной системы

Задаю язык операционной системы.

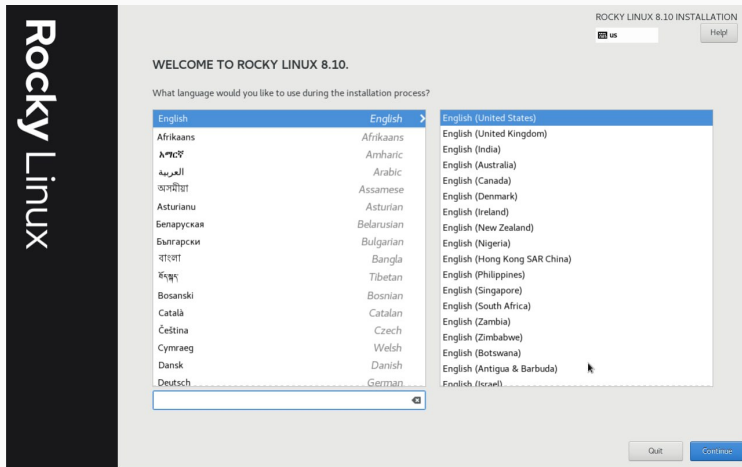


Рис. 6: Язык операционной системы

# Настройка виртуальной машины и установка операционной системы

Задаю часовой пояс операционной системы - Москва, Московское время.

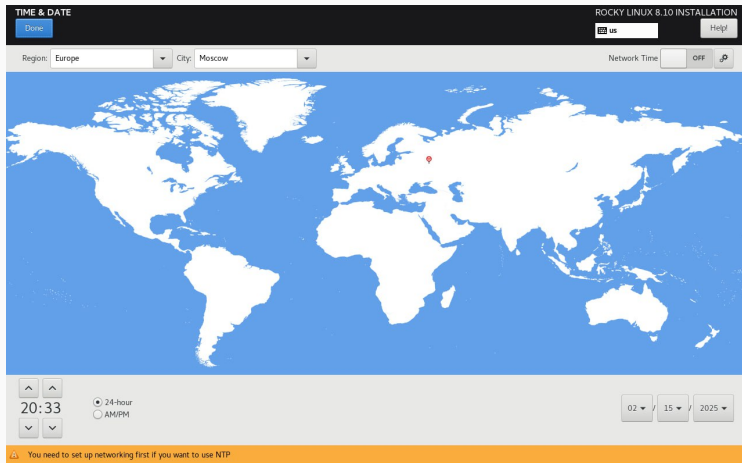
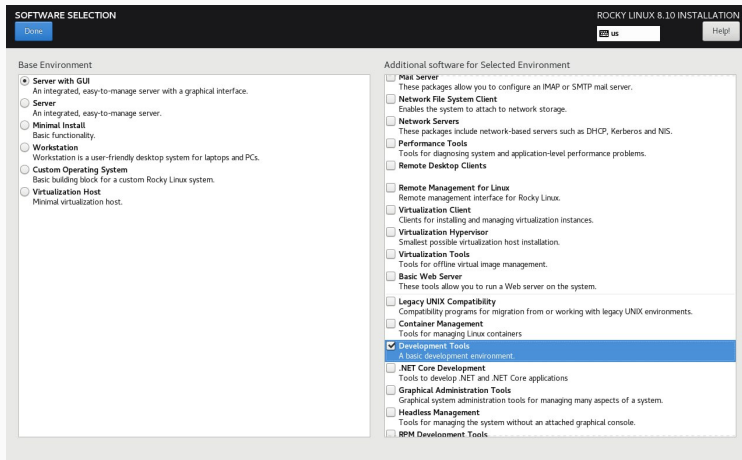


Рис. 7: Часовой пояс виртуальной машины

# Настройка виртуальной машины и установка операционной системы

В разделе выбора программ, указываю базовое окружение - Server with GUI, а в качестве дополнения - Development tools.



Настриваю клавиатуру - выбираю раскладку: русскую и английскую.

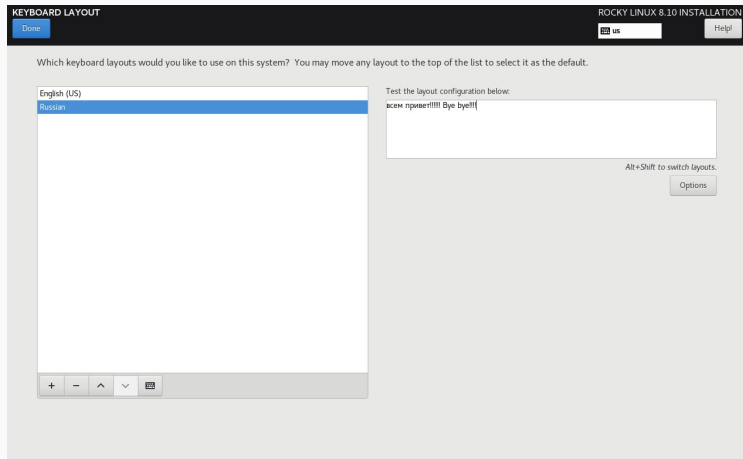


Рис. 9: Настройка клавиатуры



Отключаю KDUMP.

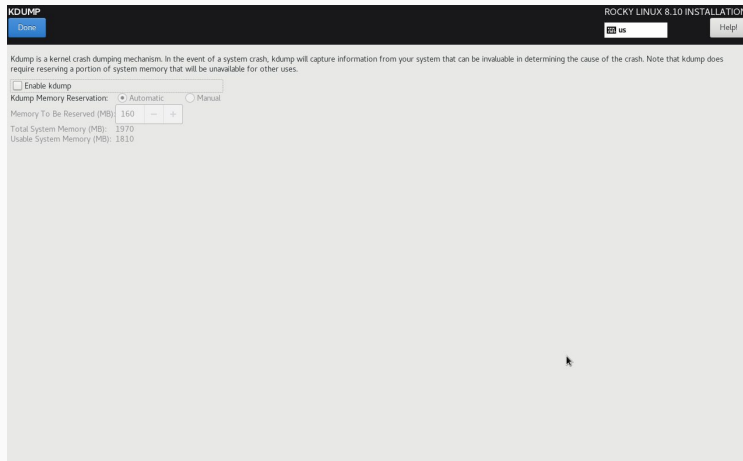


Рис. 10: Отключение KDUMP

# Настройка виртуальной машины и установка операционной системы

Включаю сетевое соединение, а в качестве имени узла указываю dmchistov.localdomain.

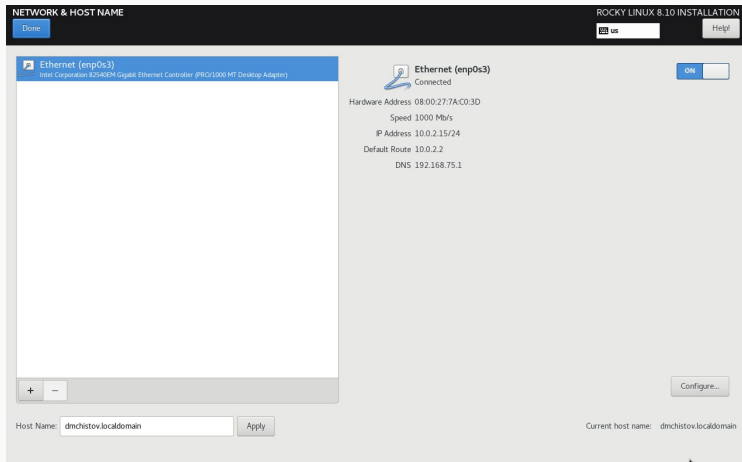
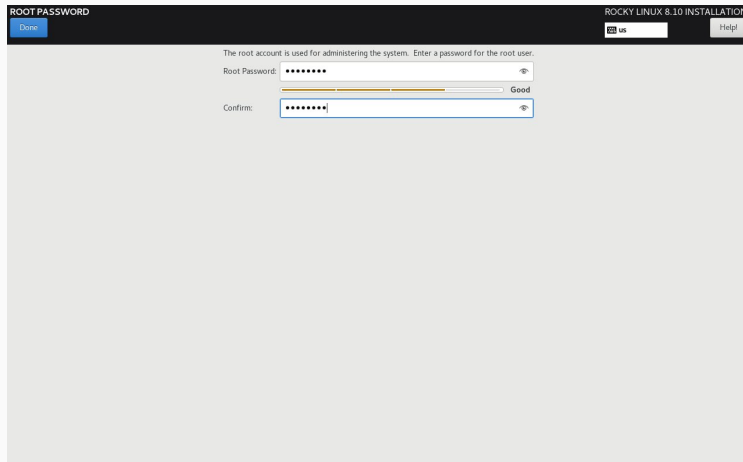


Рис. 11: Настройка сети

Устанавливаю пароль для root.



The image shows the 'ROOT PASSWORD' screen in the Rocky Linux 8.10 installation process. The title bar at the top is dark with 'ROOT PASSWORD' on the left and 'ROCKY LINUX 8.10 INSTALLATION' on the right. Below the title bar, there is a 'Done' button on the left and a 'Help' button on the right. The main content area has a light gray background. It contains the text: 'The root account is used for administering the system. Enter a password for the root user.' Below this text, there are two password input fields. The first field is labeled 'Root Password:' and contains seven asterisks. To its right is a strength indicator bar that is mostly yellow and ends with the word 'Good'. The second field is labeled 'Confirm:' and also contains seven asterisks. Both fields have an eye icon to toggle visibility. The overall layout is clean and modern.

Рис. 12: пароль для root-доступа

Создаю пользователя dmchistov и задаю ему пароль.

CREATE USER

ROCKY LINUX 8.10 INSTALLATION

Done

Help!

Full name dmchistov

User name dmchistov

☒ Make this user administrator

☒ Require a password to use this account

Password \*\*\*\*\*

Good

Confirm password \*\*\*\*\*

Advanced...

Рис. 13: Новый пользователь

Выбираю свой виртуальный жёсткий диск.

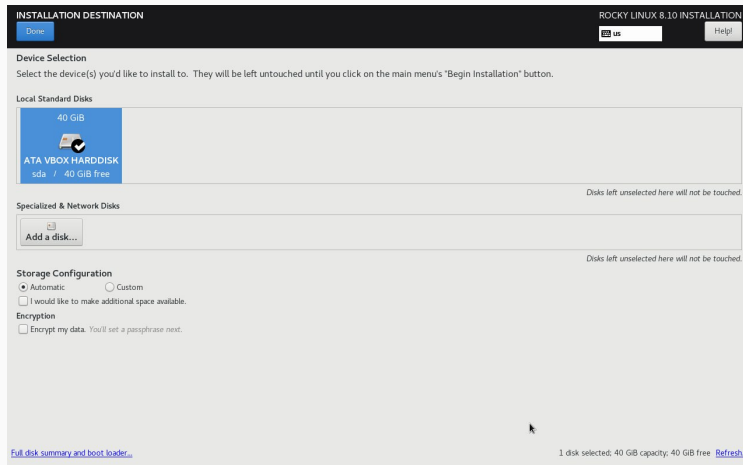


Рис. 14: Жёсткий диск для операционной системы

# Настройка виртуальной машины и установка операционной системы

Начальная настройка операционной системы завершена, перейду к установке.

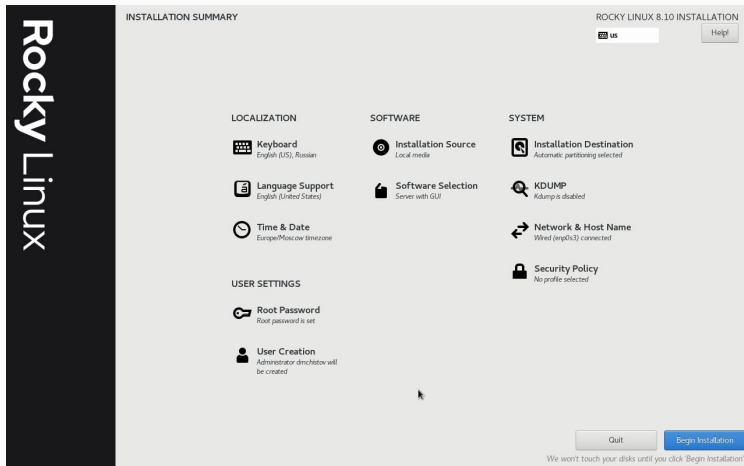


Рис. 15: Начальные настройки ОС

## Установка ОС.

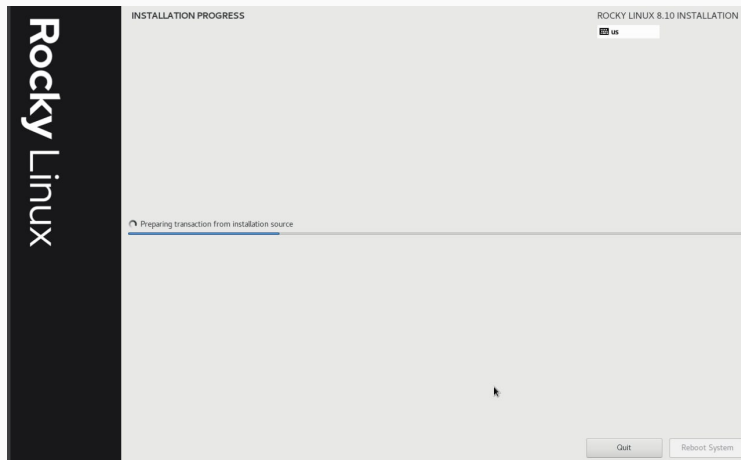


Рис. 16: Установка

Установка завершена.

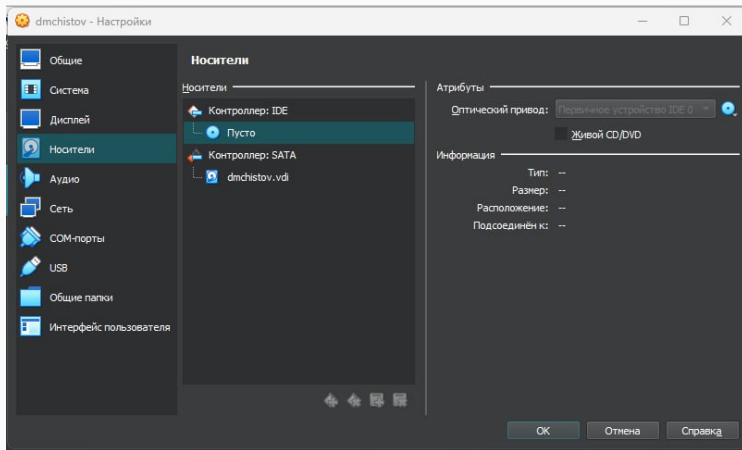


Рис. 17: Установка завершена



# Настройка виртуальной машины и установка операционной системы

Проверяю, что образ успешно автоматически исчез после установки и больше и не используется.



Подтверждаю лицензионное соглашение.

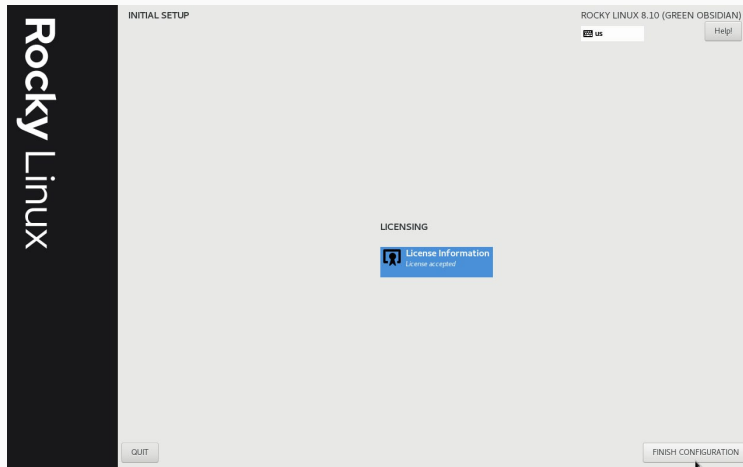


Рис. 19: Подтверждение лицензионного соглашения

После успешной установки, также подключу образ с дополнениями к моей ОС.

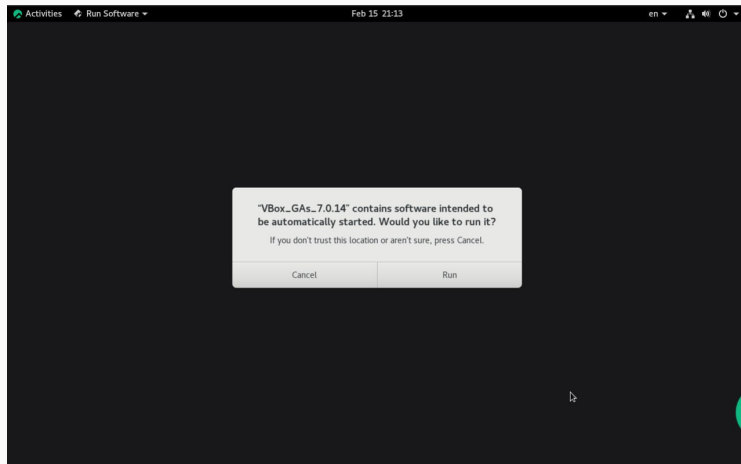


Рис. 20: Подключение дополнительных функций

## Домашнее задание

---

По заданию мне требуется узнать утилитой `dmesg` следующие вещи:

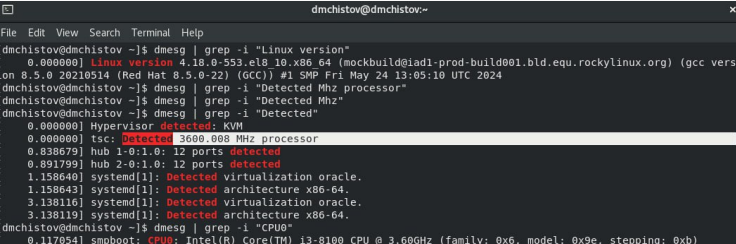
1. Версия ядра Linux (Linux version).
2. Частота процессора (Detected Mhz processor).
3. Модель процессора (CPU0).
4. Объем доступной оперативной памяти (Memory available).
5. Тип обнаруженного гипервизора (Hypervisor detected).
6. Тип файловой системы корневого раздела.
7. Последовательность монтирования файловых систем.

## Домашнее задание

Пока просто посмотрю вывод этой команды, написав `dmesg | less`.

```
dmchistov@dmchistov:~$ dmesg | less
Linux version 4.18.0-553.el8_10.x86_64 (mockbuild@iad1-prod-build001.bld.equ.rockylinux.org) (gcc version 8.5.0 20210514 (Red Hat 8.5.0-22) (GCC)) #1 SMP Fri May 24 13:05:10 UTC 2024
Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-4.18.0-553.el8_10.x86_64 root=/dev/mapper/rl_dmchistov-root ro resume=/dev/mapper/rl_dmchistov-swap rd.lvm.lv=rl_dmchistov/root rd.lvm.lv=rl_dmchistov/swap rhgb quiet
x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
x86/fpu: xstate offset[2]: 576, xstate sizes[2]: 256
x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.
signal: max sigframe size: 1776
BIOS-provided physical RAM map:
BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable
BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
BIOS-e820: [mem 0x00000000000a0000-0x00000000000fffff] reserved
BIOS-e820: [mem 0x0000000000100000-0x00000000007fffff] usable
BIOS-e820: [mem 0x00000000007fff0000-0x00000000007fffffff] ACPI data
BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
BIOS-e820: [mem 0x00000000ffff0000-0x00000000ffffffff] reserved
NX (Execute Disable) protection: active
SMBIOS 2.5 present.
DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
Hypervisor detected: KVM
kvm-clock: Using msrs 4b564d01 and 4b564d00
kvm-clock: using sched offset of 6309478735 cycles
clocksource: kvm-clock: mask: 0xffffffffffffffff max_cycles: 0x1cd42e4dffb, max_idle_ns: 881590591483 ns
tsc: Detected 3600.008 MHz processor
e820: update [mem 0x00000000-0x00000fff] usable ==> reserved
e820: remove [mem 0x000a0000-0x000fffff] usable
last_pfn = 0x7fff0 max_arch_pfn = 0x400000000
MTRR default type: uncachable
MTRR variable ranges disabled:
Disabled
x86/PAT: MTRRs disabled, skipping PAT initialization too.
CPU MTRRs all blank - virtualized system.
x86/PAT: Configuration [0-7]: WB WT UC- UC WB WT UC- UC
found SMP MP-table at [mem 0x0009ffff-0x0009ffff]
BRK [0x46801000, 0x46801fff] PGTABLE
BRK [0x46802000, 0x46802fff] PGTABLE
```

Используя `dmesg | grep -i ""` (в кавычках пишу то, что ищу), нахожу всё, что мне надо по заданию.



```
dmchistov@dmchistov:~  
File Edit View Search Terminal Help  
dmchistov@dmchistov ~]$ dmesg | grep -i "Linux version"  
0.000000] Linux version 4.18.0-553.el8_10.x86_64 (mockbuild@iad1-prod-build001.bld.equ.rockylinux.org) (gcc vers  
on 8.5.0 20210514 (Red Hat 8.5.0-22) (GCC)) #1 SMP Fri May 24 13:05:10 UTC 2024  
dmchistov@dmchistov ~]$ dmesg | grep -i "Detected Mhz processor"  
dmchistov@dmchistov ~]$ dmesg | grep -i "Detected Mhz"  
dmchistov@dmchistov ~]$ dmesg | grep -i "Detected"  
0.000000] Hypervisor detected: KVM  
0.000000] tsc: Detected 3600.008 MHz processor  
0.838679] hub 1-0:1.0: 12 ports detected  
0.891799] hub 2-0:1.0: 12 ports detected  
1.158640] systemd[1]: Detected virtualization oracle.  
1.158643] systemd[1]: Detected architecture x86-64.  
3.138116] systemd[1]: Detected virtualization oracle.  
3.138119] systemd[1]: Detected architecture x86-64.  
dmchistov@dmchistov ~]$ dmesg | grep -i "CPU0"  
0.117054] smpboot: CPU0: Intel(R) Core(TM) i3-8100 CPU @ 3.60GHz (family: 0x6, model: 0x9e, stepping: 0xb)
```

Рис. 22: Linux Version, Mhz processor, CPU0

## Домашнее задание

Также узнаю о кол-ве доступной памяти, типе гипервизора, типе файловой системы и о последовательности монтирования файловых систем.

```
0.000000] PM: Registered nosave memory: [mem 0x000a0000-0x000effff]
0.000000] PM: Registered nosave memory: [mem 0x000f0000-0x000fffff]
0.000000] Memory: 261120K/2096696K available (14339K kernel code, 5957K rwd data, 8568K rodata, 2820K init, 13792K b
ss, 138444K reserved, 0K cma-reserved)
0.014966] Freeing SMP alternatives memory: 36K
0.128194] x86/mm: Memory block size: 128MB
0.740037] Freeing initrd memory: 50468K
0.829112] Non-volatile memory driver v1.3
1.144436] Freeing unused decrypted memory: 2028K
1.144823] Freeing unused kernel image (initmem) memory: 2820K
1.149638] Freeing unused kernel image (text/rodata gap) memory: 2016K
1.149936] Freeing unused kernel image (rodata/data gap) memory: 1672K
1.778461] vmwgfx 0000:00:02.0: [drm] Legacy memory limits: VRAM = 16384 kB, FIFO = 2048 kB, surface = 507904 kB
1.778466] vmwgfx 0000:00:02.0: [drm] Maximum display memory size is 16384 kiB
dmchistov@dmchistov ~]$ dmesg | grep -i "Hypervisor detected"
0.000000] Hypervisor detected: KVM
dmchistov@dmchistov ~]$ dmesg | grep -i "sda"
2.052076] sd 2:0:0:0: [sda] 83886080 512-byte logical blocks: (42.9 GB/40.0 GiB)
2.052084] sd 2:0:0:0: [sda] Write Protect is off
2.052085] sd 2:0:0:0: [sda] Mode Sense: 00 3a 00 00
2.052092] sd 2:0:0:0: [sda] Write cache: enabled, read cache: enabled, doesn't support DPO or FUA
2.053395] sda: sda1 sda2
2.053737] sd 2:0:0:0: [sda] Attached SCSI disk
4.569939] XFS (sda1): Mounting V5 Filesystem
4.870479] XFS (sda1): Ending clean mount
dmchistov@dmchistov ~]$ dmesg | grep -i "Mount"
0.004018] Mount-cache hash table entries: 4096 (order: 3, 32768 bytes, vmalloc)
0.004023] Mountpoint-cache hash table entries: 4096 (order: 3, 32768 bytes, vmalloc)
2.531326] XFS (dm-0): Mounting V5 Filesystem
2.548943] XFS (dm-0): Ending clean mount
4.569939] XFS (sda1): Mounting V5 Filesystem
4.870479] XFS (sda1): Ending clean mount
dmchistov@dmchistov ~]$
```

Рис. 23: кол-во доступной памяти, тип гипервизора, тип файловой системы, последовательность



## Выводы

---

При выполнении данной лабораторной работы я приобрёл практические навыки установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов

## Список литературы

---

Лабораторная работы №1

Команда dmesg