

# **Отчет по лабораторной работе №1**

**по дисциплине: Информационная безопасность**

Ким Михаил Алексеевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>6</b>
3.1	Термины . . . . .	6
3.2	Окружение . . . . .	6
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
4.1	Настройка VirtualBox и ОС . . . . .	7
4.2	Подключение и настройка GitHub . . . . .	21
<b>5</b>	<b>Анализ результатов</b>	<b>28</b>
<b>6</b>	<b>Выводы</b>	<b>29</b>
	<b>Список литературы</b>	<b>30</b>

## Список иллюстраций

4.1	Загрузка ОС с официального сайта . . . . .	7
4.2	Создание виртуальной машины. 1 . . . . .	8
4.3	Создание виртуальной машины. 2 . . . . .	9
4.4	Создание виртуальной машины. 3 . . . . .	10
4.5	Создание виртуальной машины. 4 . . . . .	11
4.6	Выбор основного языка . . . . .	12
4.7	Выбор средств разработки . . . . .	13
4.8	Отключение KDUMP . . . . .	14
4.9	Выбор дополнительного языка . . . . .	15
4.10	Настройка интернет-подключения . . . . .	16
4.11	Установка . . . . .	17
4.12	Выбор дополнений гостевой ОС . . . . .	18
4.13	Установка дополнений гостевой ОС . . . . .	19
4.14	Порядок загрузки системы . . . . .	20
4.15	Просмотр интересующей информации. 1 . . . . .	21
4.16	Просмотр интересующей информации. 2 . . . . .	21
4.17	Конфигурация переменных . . . . .	22
4.18	Конфигурация ssh-ключа . . . . .	22
4.19	Конфигурация pgr-ключа. 1 . . . . .	23
4.20	Конфигурация pgr-ключа. 2 . . . . .	24
4.21	Добавление pgr-ключа на GitHub . . . . .	24
4.22	Добавленный pgr-ключ на GitHub . . . . .	25
4.23	Конфигурация переменных для быстрого доступа к GitHub . . . . .	25
4.24	Установка gh . . . . .	25
4.25	Авторизация по gh . . . . .	26
4.26	Создание репозитория на локальном хосте . . . . .	26
4.27	Создание репозитория на GitHub. 1 . . . . .	26
4.28	Создание репозитория на GitHub. 2 . . . . .	27

# 1 Цель работы

Начать знакомство с ОС Rocky Linux, вспомнить основы работы с GitHub, создать репозиторий. Вспомнить основы работ с Markdown.

## 2 Задание

1. Установить дистрибутив Linux на базе RedHat: установить операционную систему Rocky на виртуальную машину, используя средства VirtualBox.
2. Настроить систему контроля версий - Git: создать репозиторий дисциплины, связать с локальной машиной, проинициализировать начальными значениями (добавить changelog, readme, gitignore, license).

## 3 Теоретическое введение

### 3.1 Термины

Репозиторий (от англ. repository — хранилище) — место, где хранятся и поддерживаются какие-либо данные. Чаще всего данные в репозитории хранятся в виде файлов, доступных для дальнейшего распространения по сети. [1]

Контроль версий - это способ сохранять изменения с течением времени, не перезаписывая предыдущие версии. [2]

Распределенное ПО для контроля версий - каждый разработчик, работающий с репозиторием, имеет копию всего этого репозитория. [2]

### 3.2 Окружение

Rocky Linux - это корпоративная операционная система с открытым исходным кодом, разработанная таким образом, чтобы быть на 100% совместимой с Red Hat Enterprise Linux. Он находится в стадии интенсивной разработки сообществом. [3]

Git - это распределенное программное обеспечение для контроля версиями. [2]

VirtualBox - это кросс-платформенное ПО для виртуализации x86 и AMD64/Intel64 с открытым кодом для корпоративного и домашнего использования. [4]

## 4 Выполнение лабораторной работы

### 4.1 Настройка VirtualBox и ОС

1. Скачаем образ ОС с официального сайта и установим виртуальную машину (рис. 4.1, 4.2, 4.3, 4.4, 4.5).

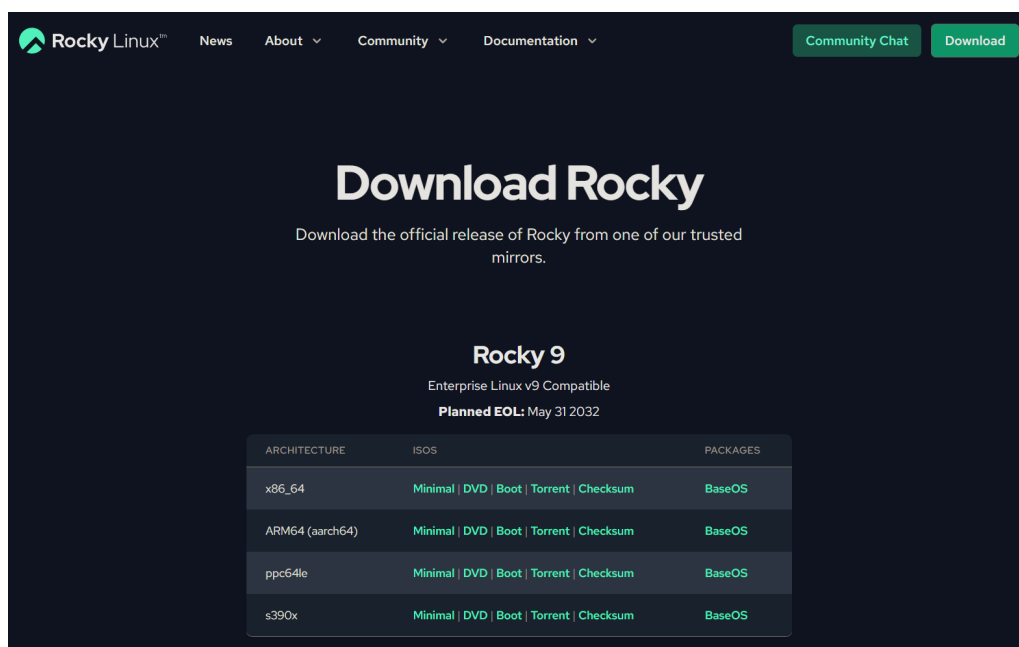


Рис. 4.1: Загрузка ОС с официального сайта

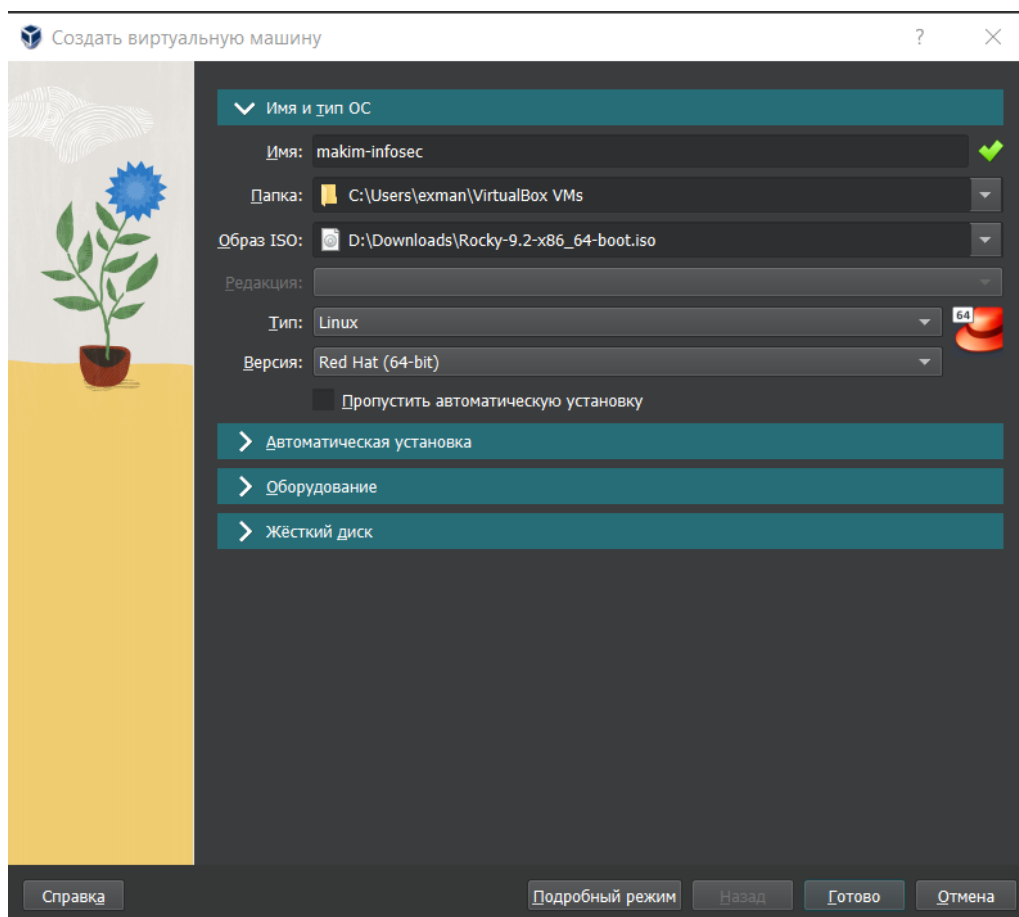


Рис. 4.2: Создание виртуальной машины. 1



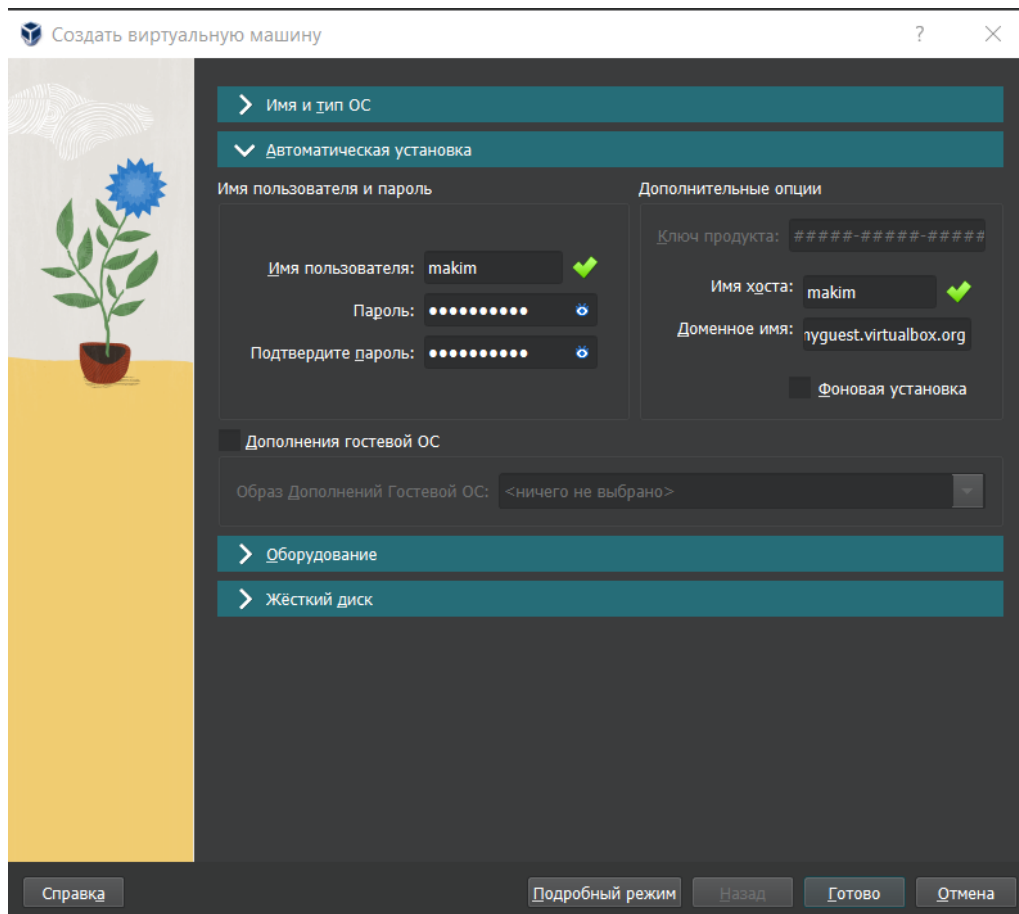


Рис. 4.3: Создание виртуальной машины. 2

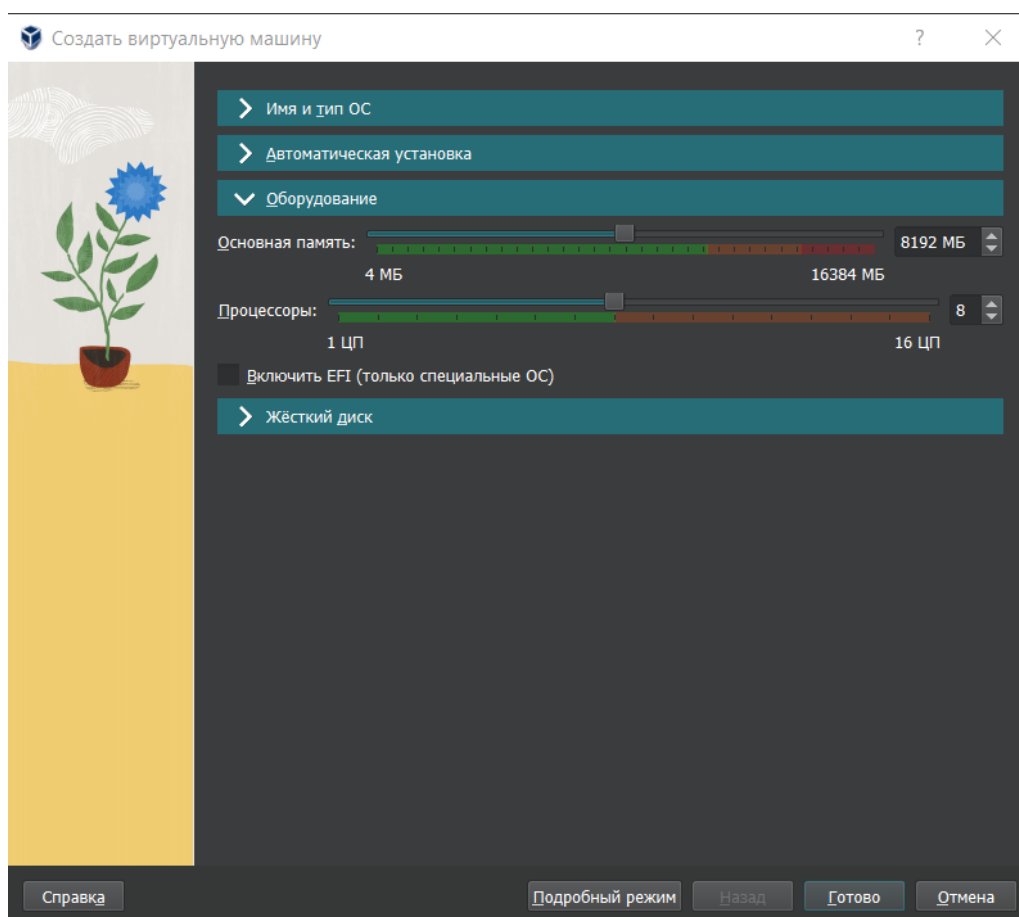


Рис. 4.4: Создание виртуальной машины. 3

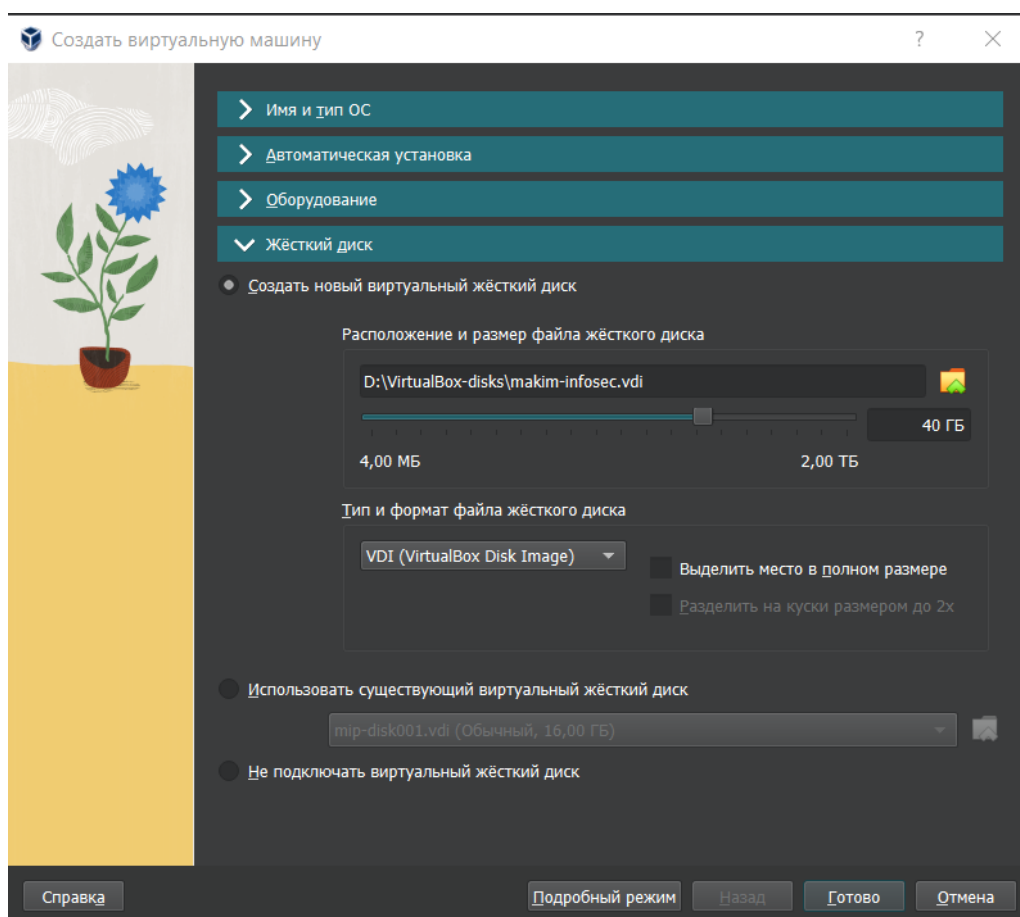


Рис. 4.5: Создание виртуальной машины. 4

2. Произведем первичные настройки ОС (рис. 4.6, 4.7, 4.8, 4.9, 4.10, 4.11).

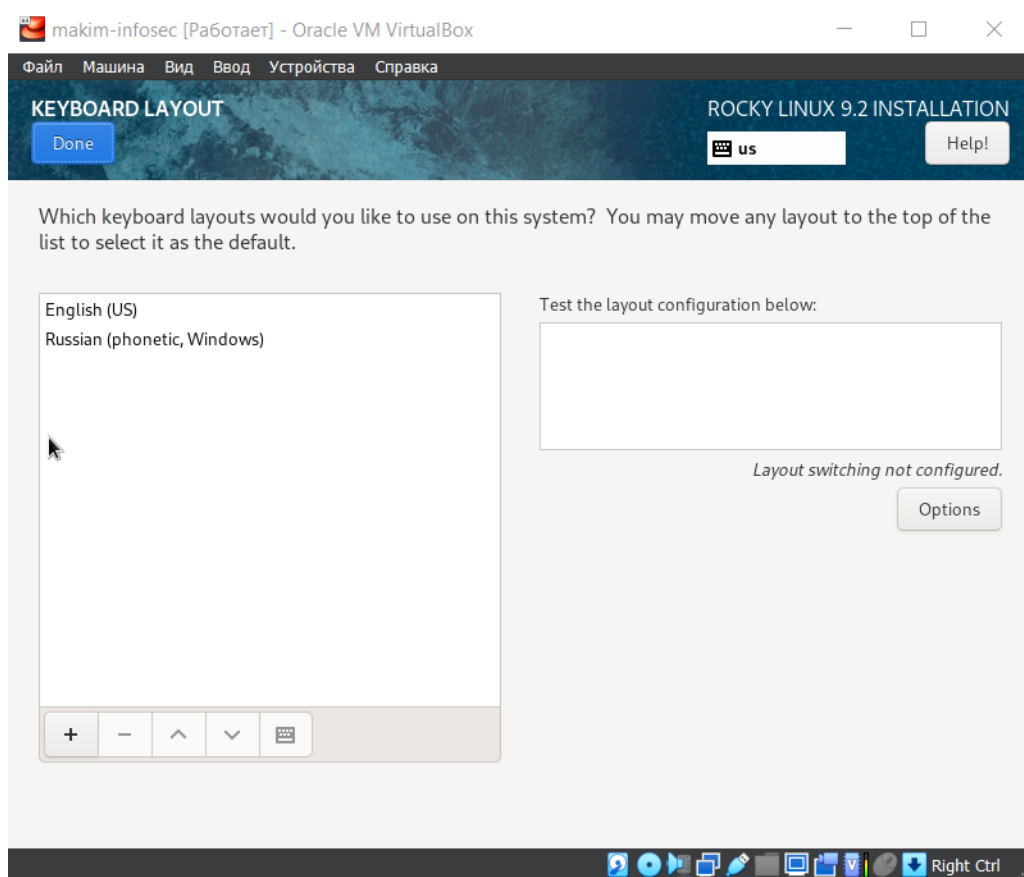


Рис. 4.6: Выбор основного языка

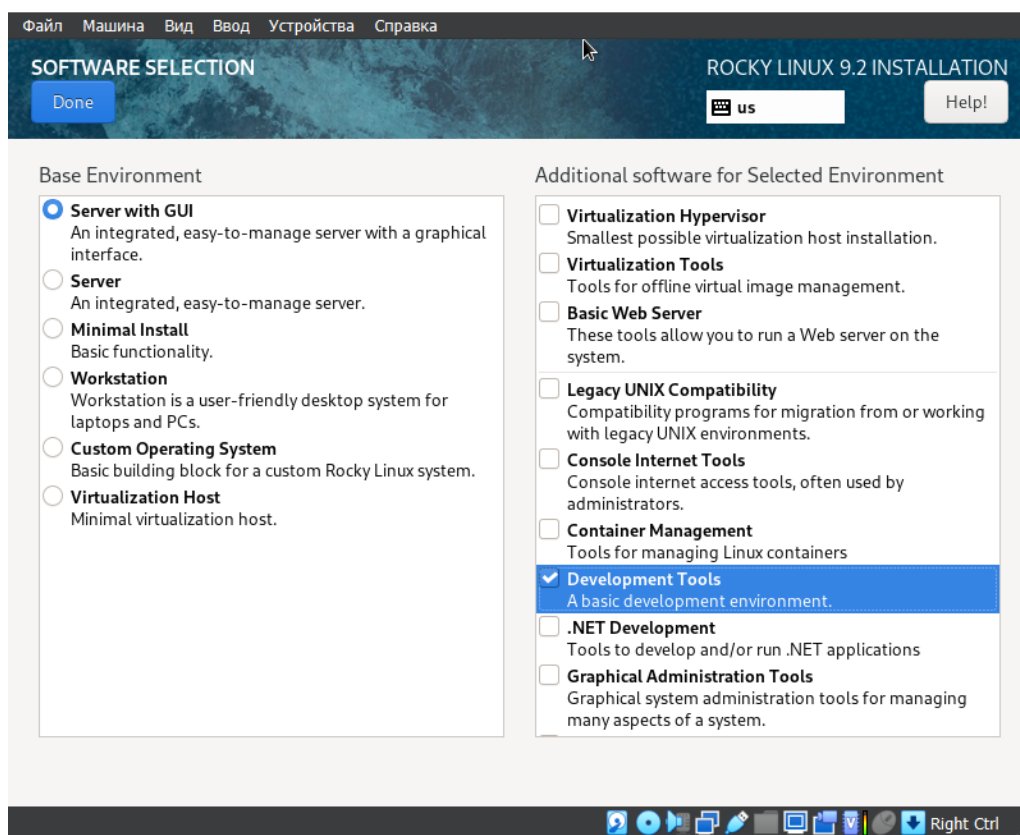


Рис. 4.7: Выбор средств разработки

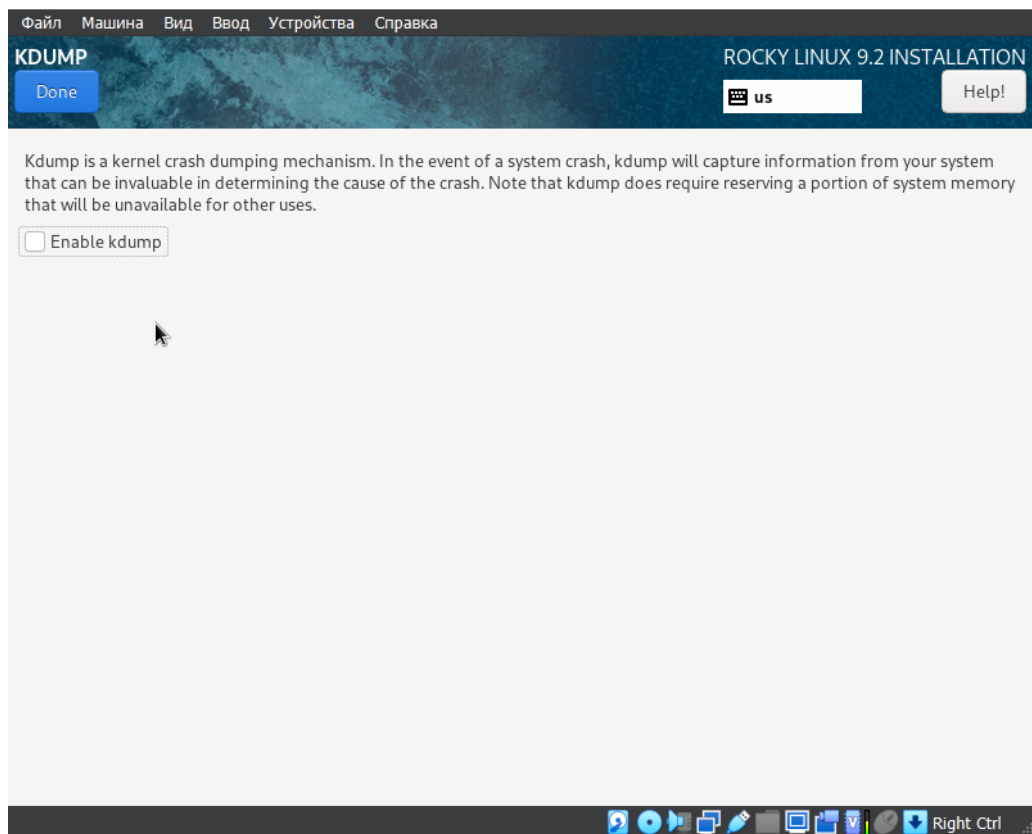


Рис. 4.8: Отключение KDUMP

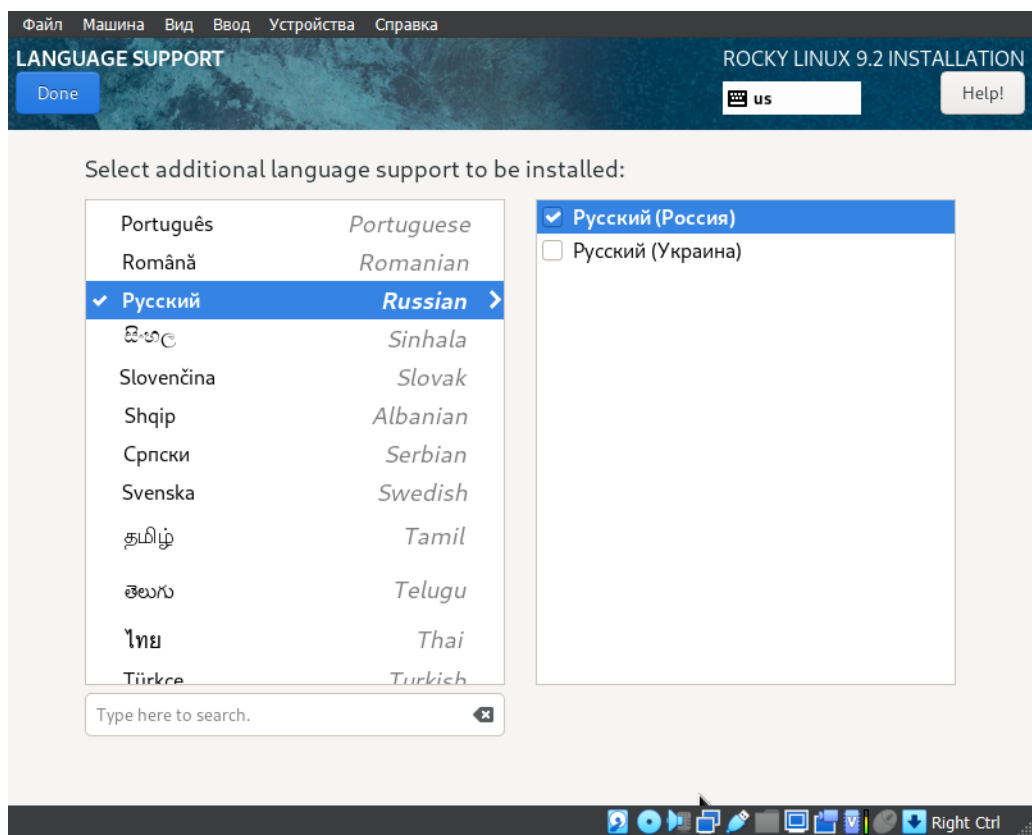


Рис. 4.9: Выбор дополнительного языка

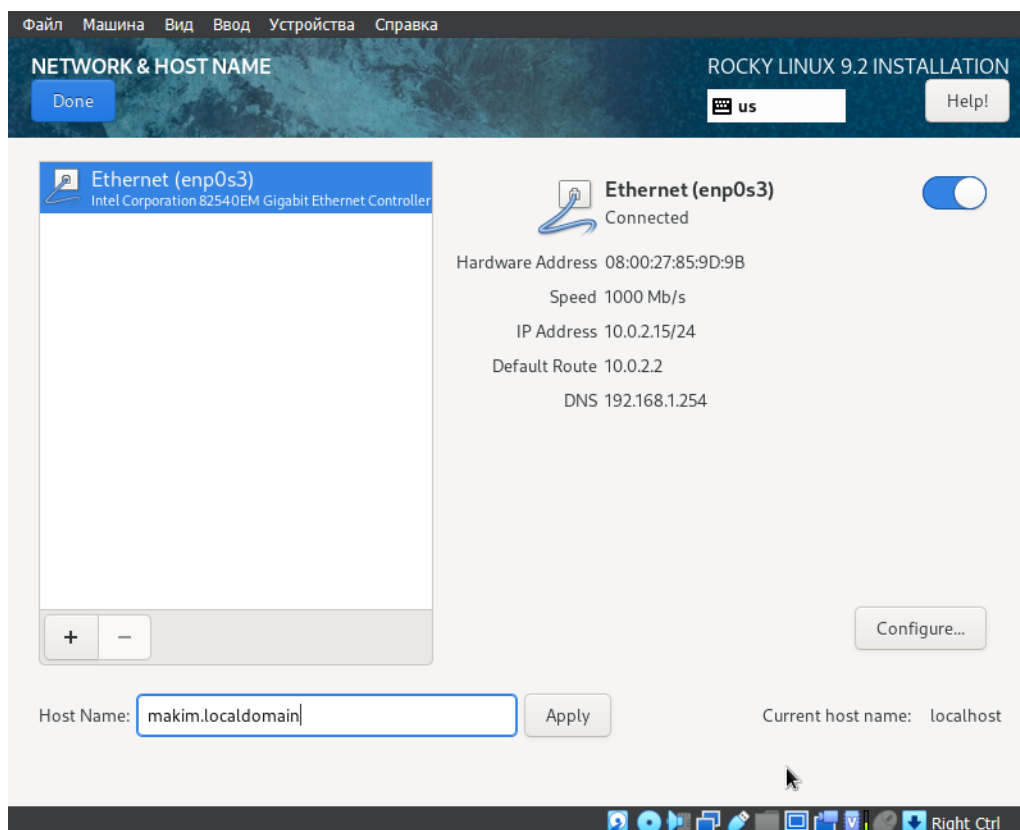


Рис. 4.10: Настройка интернет-подключения



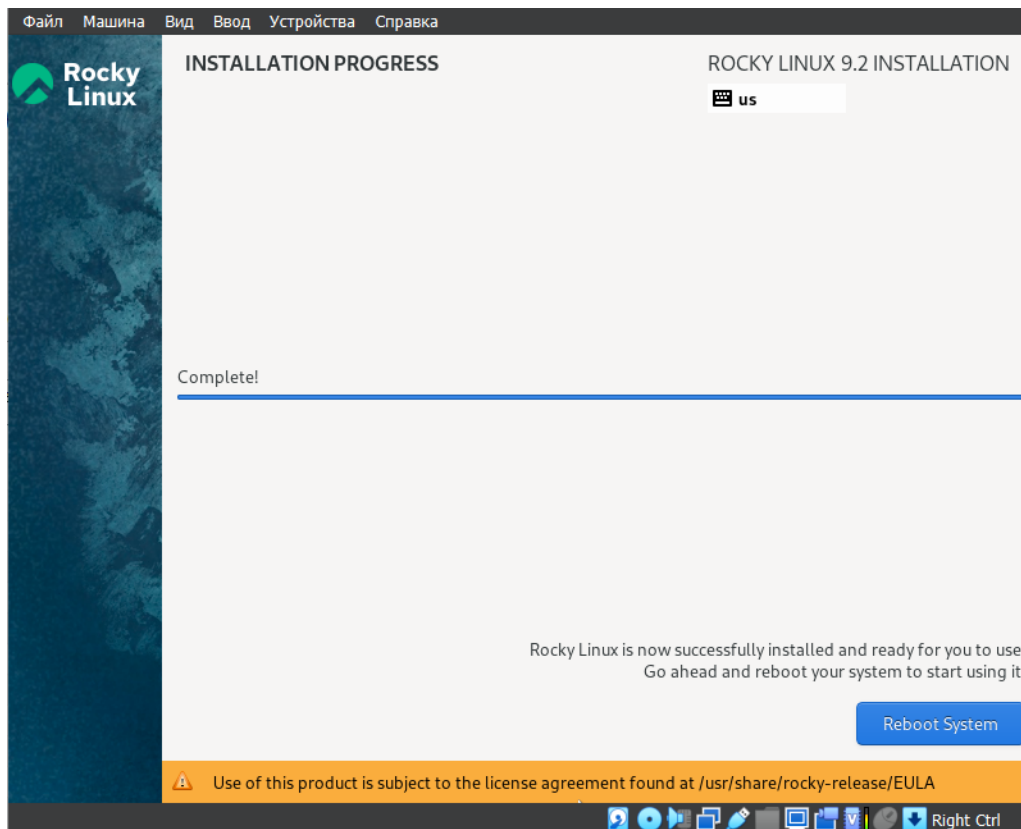


Рис. 4.11: Установка

3. Произведем установку дополнений гостевой PC (рис. 4.12, 4.13).

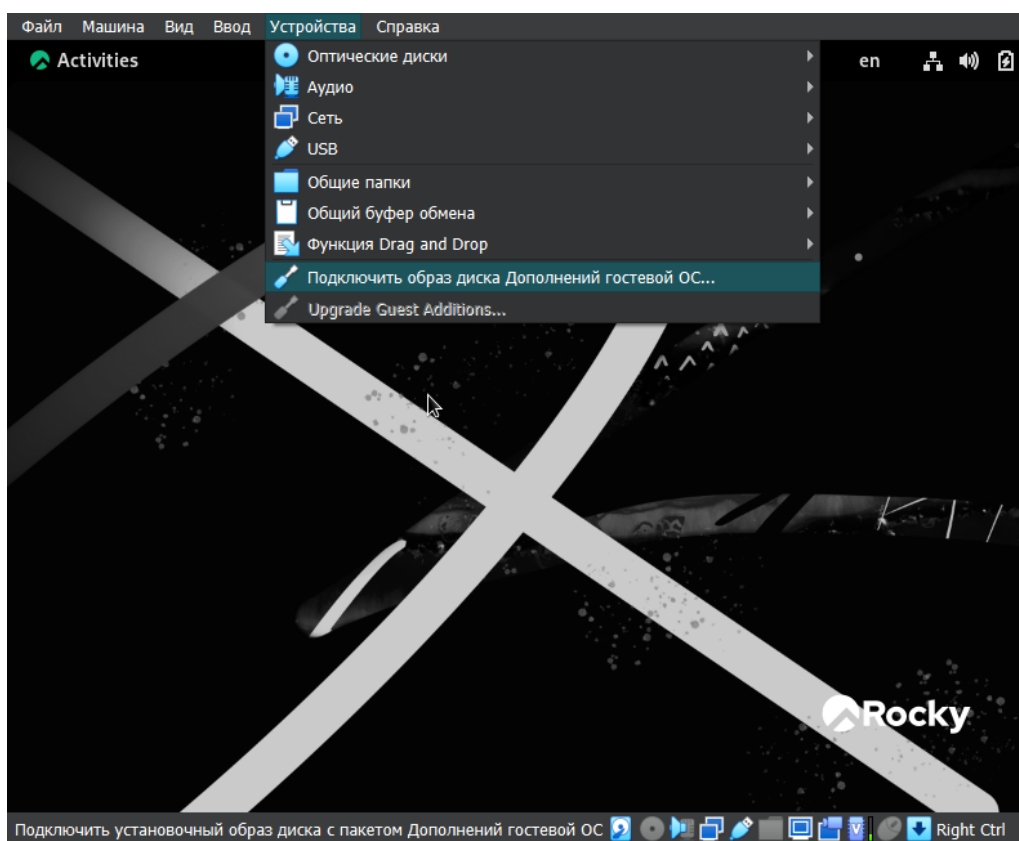


Рис. 4.12: Выбор дополнений гостевой ОС

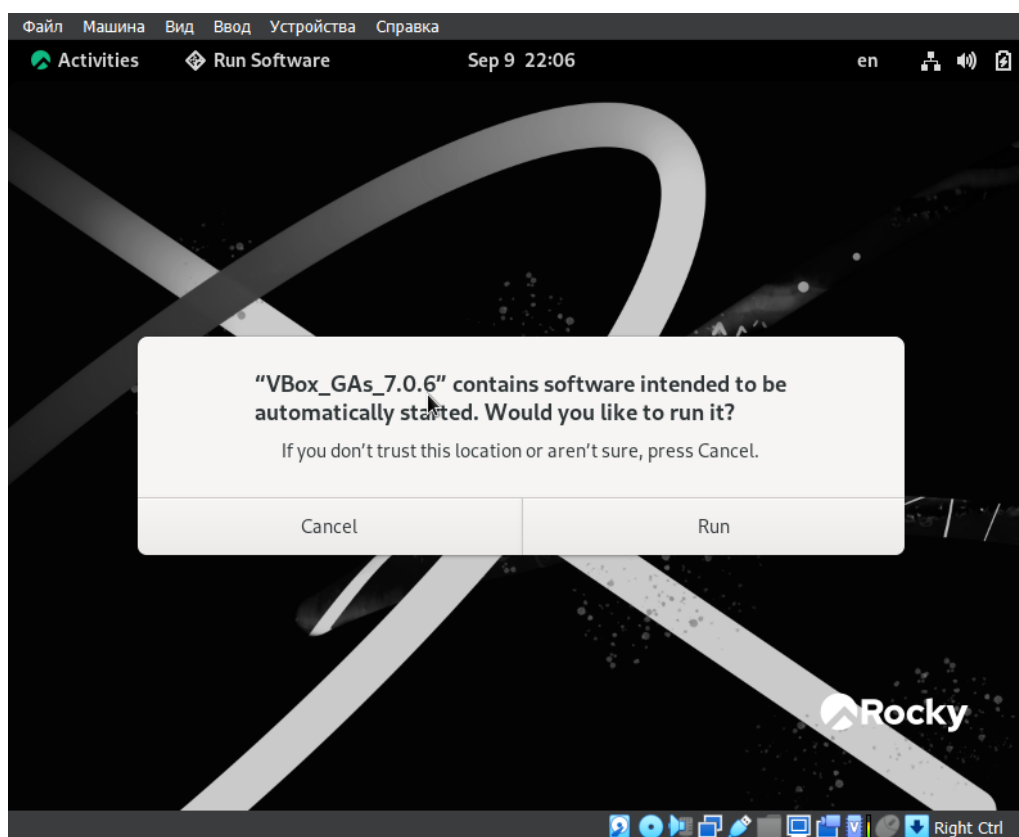


Рис. 4.13: Установка дополнений гостевой ОС

4. Просмотрим порядок загрузки системы командой `dmesg` (рис. 4.14).

```

[root@makim ~]# dmesg
0.000000 Linux version 5.14.0-284.25.1.el9_2.x86_64 (mockbuild@iad1-prod-build001.bld.equ.rockylinux.org) (gcc (GCC) 11.3.1 20221121 (Red Hat 11.3.1-4)
, GNU ld version 2.35.2-37.el9) #1 SMP PREEMPT_DYNAMIC Wed Aug 2 14:53:30 UTC 2023
0.000000 The list of certified hardware and cloud instances for Enterprise Linux 9 can be viewed at the Red Hat Ecosystem Catalog, https://catalog.redh
at.com.
0.000000 Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-5.14.0-284.25.1.el9_2.x86_64 root=/dev/mapper/rl-root ro resume=/dev/mapper/rl-swap rd.lvm.lv=rl
/root rd.lvm.lv=rl/swap rhgb quiet
0.000000 [Firmware bug]: tsc doesn't count with PG frequency!
0.000000 x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
0.000000 x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
0.000000 x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
0.000000 x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
0.000000 x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.
0.000000 signal: max sigframe size: 1776
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 0x000000000009f000-0x000000000009ffff] reserved
0.000000 BIOS-e820: [mem 0x0000000000100000-0x00000000000dffff] usable
0.000000 BIOS-e820: [mem 0x000000000dffff0000-0x00000000dffffffffff] ACPI data
0.000000 BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
0.000000 BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
0.000000 BIOS-e820: [mem 0x00000000fffc0000-0x00000000fffc0fff] reserved
0.000000 BIOS-e820: [mem 0x0000000100000000-0x0000000021ffffff] usable
0.000000 NX (Execute Disable) protection: active
0.000000 SMBIOS 2.5 present.
0.000000 DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
0.000000 Hypervisor detected: KVM
0.000000 kvm-clock: Using mrs 4b564d01 and 4b564d00
0.000000 kvm-clock: using sched offset of 4761805024 cycles
0.000000 clocksource: kvm-clock: mask: 0xffffffffffffff max_cycles: 0x1cd42e4dffb, max_idle_ns: 881590951483 ns
0.000121 tsc: Detected 3194.002 MHz processor
0.001273 e820: update [mem 0x00000000-0x000000fff] usable ==> reserved
0.001279 e820: remove [mem 0x000a0000-0x0000ffff] usable
0.001285 last_pfn = 0x220000 max_arch_pfn = 0x400000000
0.001313 Disabling
0.001315 x86/PAT: MTRRs disabled, skipping PAT initialization too.
0.001317 CPU MTRRs all blank - virtualized system.
0.001322 x86/PAT: Configuration [0-7]: WB WT UC- UC WB WT UC- UC
0.001327 last_pfn = 0xdffff max_arch_pfn = 0x400000000
0.001354 Found SMP MP-table at [mem 0x000ffff0-0x0009ffff]
0.001673 RMOBS: [mem 0x314d000-0x314b6fff]
0.001677 ACPI: Early table checksum verification disabled
0.001681 ACPI: RSDP 0x00000000000E0000 000024 (v02 VBOX )
0.001685 ACPI: XSDT 0x000000000DFF0030 00003C (v01 VBOX VBOXXSDT 00000001 ASL 00000061)
0.001691 ACPI: FACP 0x000000000DFF00F0 0000F4 (v04 VBOX VBOXFACP 00000001 ASL 00000061)
0.001696 ACPI: DSDT 0x000000000DFF0640 002353 (v02 VBOX VBOXBIOS 00000002 INTL 20100528)
0.001698 ACPI: FACS 0x00000000DFF0200 000040 (v02 VBOX VBOXBIOS 00000002 INTL 20100528)
0.001699 ACPI: FACS 0x00000000DFF0200 000040
0.001701 ACPI: APIC 0x000000000DFF0240 00008C (v02 VBOX VBOXAPIC 00000001 ASL 00000061)
0.001703 ACPI: SSDT 0x00000000DFF02D0 00036C (v01 VBOX VBOXCPU 00000002 INTL 20100528)
0.001704 ACPI: Reserving FACP table memory at [mem 0xdffff00f0-0xdffff01e3]
0.001705 ACPI: Reserving DSDT table memory at [mem 0xdffff0640-0xdffff2992]
0.001706 ACPI: Reserving FACS table memory at [mem 0xdffff0200-0xdffff023f]
0.001706 ACPI: Reserving FACS table memory at [mem 0xdffff0200-0xdffff023f]
0.001916 ACPI: Reserving APIC table memory at [mem 0xdffff0240-0xdffff02cb]
0.001917 ACPI: Reserving SSDT table memory at [mem 0xdffff02d0-0xdffff063b]
0.002100 No NUMA configuration found
0.002182 Faking a node at [mem 0x0000000000000000-0x0000000021ffffff]
0.002189 NODE_DATA(0) allocated [mem 0x21fffd1000-0x21fffbfff]
0.002447 Zone ranges:
0.002447 DMA [mem 0x00000000000001000-0x000000000000ffff]
0.002449 DMA32 [mem 0x0000000001000000-0x000000000000ffff]
0.002450 Normal [mem 0x0000000100000000-0x0000000021ffffff]
0.002451 Device empty
0.002452 Movable zone start for each node
0.002453 Early memory node ranges
0.002453 node 0: [mem 0x00000000000001000-0x0000000000009eff]
0.002455 node 0: [mem 0x0000000001000000-0x000000000dffff]
0.002455 node 0: [mem 0x0000000100000000-0x0000000021ffffff]
0.002457 Initmem setup node 0 [mem 0x00000000000001000-0x0000000021ffffff]
0.002508 On node 0, zone DMA: 1 pages in unavailable ranges
0.002536 On node 0, zone DMA: 97 pages in unavailable ranges
0.010602 On node 0, zone Normal: 16 pages in unavailable ranges

```

Рис. 4.14: Порядок загрузки системы

5. Просмотрим интересующую информацию командой `dmesg | grep -i <to-find>` (рис. 4.15, 4.16).

```

root@makim ~]# dmesg | grep -i "Linux version"
[ 0.000000] Linux version 5.14.0-284.25.1.el9_2.x86_64 (mockbuild@fadi-prod-build001.bld.equ.rockylinux.org) (gcc (GCC) 11.3.1 20221121 (Red Hat 11.3.1-4)
GNU ld version 2.35.2-37.el9) #1 SMP PREEMPT_DYNAMIC Wed Aug 2 14:53:30 UTC 2023
root@makim ~]# dmesg | grep -i "Mhz processor"
[ 0.000012] tsc: Detected 3194.002 MHz processor
root@makim ~]# dmesg | grep -i "CPU0"
[ 0.178539] smpboot: CPU0: AMD Ryzen 7 6800H with Radeon Graphics (family: 0x19, model: 0x44, stepping: 0x1)
root@makim ~]# dmesg | grep -i "Memory:"
[ 0.011028] PM: hibernation: Registered nosave memory: [mem 0x00000000-0x00000fff]
[ 0.011030] PM: hibernation: Registered nosave memory: [mem 0x0000f000-0x0000ffff]
[ 0.011030] PM: hibernation: Registered nosave memory: [mem 0x000a0000-0x000effff]
[ 0.011031] PM: hibernation: Registered nosave memory: [mem 0x000f0000-0x000fffff]
[ 0.011032] PM: hibernation: Registered nosave memory: [mem 0xdfff0000-0xdfffffff]
[ 0.011032] PM: hibernation: Registered nosave memory: [mem 0xe0000000-0xfefbffff]
[ 0.011033] PM: hibernation: Registered nosave memory: [mem 0xfec00000-0xfec0ffff]
[ 0.011033] PM: hibernation: Registered nosave memory: [mem 0xfec01000-0xfedfffff]
[ 0.011033] PM: hibernation: Registered nosave memory: [mem 0xfef00000-0xfef0ffff]
[ 0.011034] PM: hibernation: Registered nosave memory: [mem 0xfef01000-0xfefbffff]
[ 0.011034] PM: hibernation: Registered nosave memory: [mem 0xffffc000-0xffffffff]
[ 0.040821] Memory: 3632208K/8388152K available (14342K kernel code, 5536K rdata, 10180K rodata, 2792K init, 7524K bss, 316360K reserved, 0K cma-reserved)
[ 0.075478] Freeing SMP alternatives memory: 36K
[ 0.747733] Freeing initrd memory: 54824K
[ 0.878382] Freeing unused decrypted memory: 2036K
[ 0.878869] Freeing unused kernel image (initmem) memory: 2792K
[ 0.880210] Freeing unused kernel image (text/rodata gap) memory: 2040K
[ 0.880255] Freeing unused kernel image (rodata/data gap) memory: 60K
root@makim ~]# dmesg | grep -i "Hypervisor"
[ 0.000000] Hypervisor detected: KVM

```

Рис. 4.15: Просмотр интересующей информации. 1

```

root@makim ~]# dmesg | grep -i "Filesystem"
[ 2.457457] XFS (dm-0): Mounting V5 Filesystem
[ 4.007147] XFS (sda1): Mounting V5 Filesystem
root@makim ~]# dmesg | grep -i "Mounting"
[ 2.457457] XFS (dm-0): Mounting V5 Filesystem
[ 3.112806] systemd[1]: Mounting Huge Pages File System...
[ 3.114057] systemd[1]: Mounting POSIX Message Queue File System...
[ 3.115341] systemd[1]: Mounting Kernel Debug File System...
[ 3.116788] systemd[1]: Mounting Kernel Trace File System...
[ 4.007147] XFS (sda1): Mounting V5 Filesystem

```

Рис. 4.16: Просмотр интересующей информации. 2

## 4.2 Подключение и настройка GitHub

1. Конфигурируем git и создаем ключи (рис. 4.17, 4.18, 4.19, 4.20, 4.21, 4.22, 4.23).

```
[makim@makim ~]$ git config --global user.name "Michael Kim"
[makim@makim ~]$ git config --global user.email "1032201664@pfur.ru"
[makim@makim ~]$ mkdir -p ~/work/study/2023-2024/"Информационная безопасность"
[makim@makim ~]$ cd ~/work/study/2023-2024/"Информационная безопасность"
[makim@makim Информационная безопасность]$ git congfig --clobal core.quoteparh false
git: 'congfig' is not a git command. See 'git --help'.

The most similar command is
    config
[makim@makim Информационная безопасность]$ git config --global core.quoteparh false
[makim@makim Информационная безопасность]$ git config --global init.defaultBranch master
[makim@makim Информационная безопасность]$ git config --global core.autocrlf input
[makim@makim Информационная безопасность]$ git config --global core.safecrlf warn
```

Рис. 4.17: Конфигурация переменных

```
[makim@makim Информационная безопасность]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/makim/.ssh/id_rsa):
Created directory '/home/makim/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match. Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/makim/.ssh/id_rsa
Your public key has been saved in /home/makim/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:xoKVxMZaLakWd36WmLjjY2JDMt0Lm70py8QU90B+5To makim@makim.localdomain
The key's randomart image is:
+----[RSA 4096]-----+
|      .o.o.      |
|      o..0+o     |
|      . +B+=.o .  |
|      o+*.o+ +    |
|      .o. E.So    |
|    o+ o o+      |
|      o* o .      |
|    000=.,+      |
|      ==+=.      |
+-----[SHA256]-----+
```

Рис. 4.18: Конфигурация ssh-ключа

```

[ makim@makim Информационная безопасность]$ gpg --full-generate-key
gpg (GnuPG) 2.3.3; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory '/home/makim/.gnupg' created
gpg: keybox '/home/makim/.gnupg/pubring.kbx' created
Please select what kind of key you want:
  (1) RSA and RSA (default)
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid.
    0 = key does not expire
    <n>  = key expires in n days
    <n>w  = key expires in n weeks
    <n>m  = key expires in n months
    <n>y  = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: Michael

```

Рис. 4.19: Конфигурация ргр-ключа. 1

```
[makim@makim ~]$ gpg --list-secret-keys --keyid-format LONG
/home/makim/.gnupg/pubring.kbx
-----
sec   rsa4096/792E0821851DFEA3 2023-09-09 [SC]
      824A484C927A5190302E6E65792E0821851DFEA3
uid           [ultimate] Michael (first_try) <exmanka@gmail.com>
ssb   rsa4096/F7EFF3840939E7BD 2023-09-09 [E]

[makim@makim ~]$ gpg --armor --export 792E0821851DFEA3 | xclip -sel clip
bash: xclip: command not found...
[makim@makim ~]$ gpg --armor --export 792E0821851DFEA3
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGT8ynsBEACcP/S+PkHh0ffhbnetLurkB0QBL5g/pstw87V+eXYog6Wc+Lik
/ZkBmjPjHXgy0bmJTQNYk47aiKiG9vFAP9SP0u2WuhtTsbHzth+XRDThSfLGRuGbp
O+Xae9t7ScaNpgmb7NrvQxI2iR2kdL04BX9PDvY+1aixsXciM5Bmg2t3l6RnJxgr
CQudEmXwzNu0k8HLkrESLLWTTKjB/5JV0N3HVQ01yrf43zVVqheEBoXD5PQiQH3m
/Gy3HdX8g8UToA+hQNEzhkHZ1C8ajiV/opFzbV741/sqovfqbv6EQdKeqtvtM+n
phkJm/2IEfspDbaqs9cf47XAsileV9yqHrIfLBpPmbLW9g6DtNmfyNeRm90uwdpK
ONesb8Ze1lKRiF8oQX68gNakmaVstHX1fKH1dJL7lIX68dWnN22e7P60b8lcXAAI
-----
```

Рис. 4.20: Конфигурация ргр-ключа. 2

### Add new GPG key

Title

Key

```
sq0lD6iJLzumqdd0hKBNCaGZsg4URz7/fn/8f0zPUIFI+TjzFEeo1Fb17Yvy/H1E
V7WW7FzTgtWCt/pgmnPzFr+HVprqeVTgoX+LU9mMhX3zn/O7wLR3pdFgBj9l8004
Ul0dLMIGg/L08xew042v2uMF732c5NHEYqhK74sjROWhN7kzPXMfOB7yg2QaqSvD
NQrA07RSSsyrbtv3B8Umtpl0LGg/60yGXDmKbQhDEO2uTxLrXsLLidLUZUGuGSFPB
anLsJ3TSC9yzlGEauj6CKsfsum0ybtOlghNaqdoYF1YogBQHUh1DkMg33+INeVO
ffR7DYW75Q==
=W/VD

-----END PGP PUBLIC KEY BLOCK-----
```

Add GPG key

Рис. 4.21: Добавление ргр-ключа на GitHub



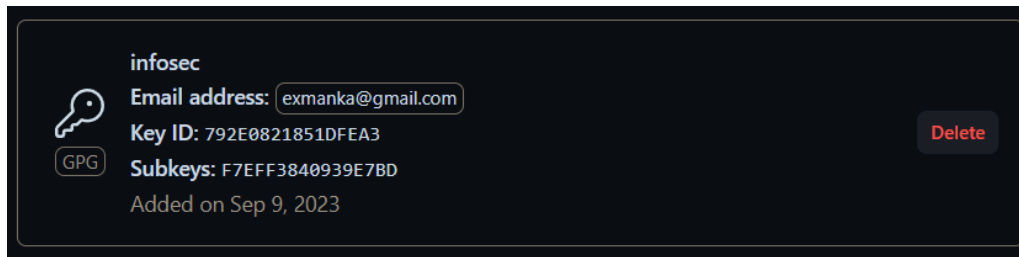


Рис. 4.22: Добавленный pgp-ключ на GitHub

```
[makim@makim ~]$ git config --global user.signingkey 792E0821851DFEA3
[makim@makim ~]$ git config --global commit.gpgsign true
[makim@makim ~]$ git config --global gpg.program $(which gpg2)
[makim@makim ~]$
```

Рис. 4.23: Конфигурация переменных для быстрого доступа к GitHub

## 2. Установка gh и подключение (рис. 4.24, 4.25, 4.26, 4.27, 4.28).

```
Error: Unable to find a match: gh
[makim@makim ~]$ sudo dnf config-manager --add-repo https://cli.github.com/packages/rpm/gh-cli.repo
[sudo] password for makim:
Adding repo from: https://cli.github.com/packages/rpm/gh-cli.repo
[makim@makim ~]$ sudo dnf install gh
packages for the GitHub CLI                               7.3 kB/s | 2.6 kB    00:00
Dependencies resolved.
=====
Package      Architecture  Version      Repository      Size
=====
Installing:
gh           x86_64        2.34.0-1     gh-cli          11 M
Transaction Summary
=====
Install 1 Package
```

Рис. 4.24: Установка gh

```

[makim@makim ~]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? SSH
? Upload your SSH public key to your GitHub account? /home/makim/.ssh/id_rsa.pub
? Title for your SSH key: infosec
? How would you like to authenticate GitHub CLI? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'admin:public_key'.
? Paste your authentication token: *****
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ Uploaded the SSH key to your GitHub account: /home/makim/.ssh/id_rsa.pub
✓ Logged in as exmanka

```

Рис. 4.25: Авторизация по gh

```

[makim@makim ~]$ mkdir -p ~/work/study/2023-2024/"Информационная безопасность"
[makim@makim ~]$ cd ~/work/study/2023-2024/"Информационная безопасность"
[makim@makim Информационная безопасность]$ gh repo create study_2023-2024_infosec --template=yamadharma/course-directory-studenttemplate --public
GraphQL: Could not resolve to a Repository with the name 'yamadharma/course-directory-studenttemplate'. (repository)
[makim@makim Информационная безопасность]$ gh repo create study_2023-2024_infosec --template=yamadharma/course-directory-student-template --public
✓ Created repository exmanka/study_2023-2024_infosec on GitHub
[makim@makim Информационная безопасность]$ git clone --recursive git@github.com:owner>/study_2022-2023_infosec.git infosec
bash: owner: No such file or directory
[makim@makim Информационная безопасность]$ git clone --recursive git@github.com:exmanka/study_2023-2024_infosec.git infosec
Cloning into 'infosec'...
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TUJJhbpZisF/zLDA0zPMSvHdKr4UvCOqU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

```

Рис. 4.26: Создание репозитория на локальном хосте

```

[makim@makim Информационная безопасность]$ cd infosec
[makim@makim infosec]$ rm package.json
[makim@makim infosec]$ echo "infosec" > COURSE
[makim@makim infosec]$ make
[makim@makim infosec]$ git add -A
[makim@makim infosec]$ git commit -am 'feat(main): make course structure'
[master e238360] feat(main): make course structure
151 files changed, 41045 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab1/presentation/Makefile
create mode 100644 labs/lab1/presentation/image/kulyabov.jpg
create mode 100644 labs/lab1/presentation/presentation.md

```

Рис. 4.27: Создание репозитория на GitHub. 1

```
[makim@makim infosec]$ git push
Enumerating objects: 37, done.
Counting objects: 100% (37/37), done.
Delta compression using up to 8 threads
Compressing objects: 100% (29/29), done.
Writing objects: 100% (35/35), 342.77 KiB | 2.50 MiB/s, done.
Total 35 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:exmanka/study_2023-2024_infosec.git
 3977bb0..e238360 master -> master
```

Рис. 4.28: Создание репозитория на GitHub. 2

## 5 Анализ результатов

Работа выполнена без каких-либо проблем. ОС Rocky Linux показала себя отлично. Особенно хотелось бы выделить скорость работы и графический интерфейс. Также порадовало наличие вкладок в терминале «из коробки».

## 6 Выводы

Начал знакомство с ОС Rocky Linux, вспомнил основы работы с GitHub, создал репозитрий. Вспомнил основы работ с Markdown.

## Список литературы

1. Репозиторий [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%B9%7D>.
2. Git-Guides [Электронный ресурс]. URL: <https://github.com/git-guides>.
3. Документация Rocky Linux [Электронный ресурс]. URL: <https://docs.rockylinux.org/>.
4. Установка Rocky на VirtualBox [Электронный ресурс]. URL: <https://www.virtualbox.org/>.