

# **Лабораторная работа № 1**

**Установка и конфигурация операционной системы на  
виртуальную машину и подготовка репозитория**

Тарусов Артём Сергеевич

# Содержание

<b>Цель работы</b>	<b>4</b>
<b>Задание</b>	<b>5</b>
<b>Теоретическое введение</b>	<b>6</b>
<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>Выводы</b>	<b>15</b>
<b>Список литературы</b>	<b>16</b>

## Список иллюстраций

1	Окно создания виртуальной машины . . . . .	7
2	Окно настройки ОС . . . . .	8
3	Версия ядра Linux . . . . .	8
4	Частота процессора . . . . .	8
5	Модель процессора . . . . .	8
6	Объем доступной оперативной памяти . . . . .	9
7	Тип обнаруженного гипервизора . . . . .	9
8	Тип файловой системы корневого раздела . . . . .	9
9	Последовательность монтирования файловых систем . . . . .	10
10	Создание рабочей директории . . . . .	10
11	Базовая настройка git . . . . .	10
12	Создание ключа SSH по алгоритму rsa . . . . .	11
13	Создание ключа SSH по алгоритму ed25519 . . . . .	11
14	Создание ключа GPG . . . . .	11
15	Создание ключа GPG . . . . .	12
16	Добавление ключа GPG в Github . . . . .	13
17	Настройка автоматических подписей коммитов git . . . . .	13
18	Настройка gh . . . . .	14
19	Создание репозитория . . . . .	14
20	Клонирование репозитория . . . . .	14
21	Настройка каталога курса . . . . .	14

# Цель работы

Целью данной работы является приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов, а также изучение средств контроля версий и получение навыков работы с git.

## **Задание**

1. Установить и настроить ОС Linux на виртуальную машину.
2. Подготовить репозиторий для дальнейшей работы.

# Теоретическое введение

- Операционная система — это комплекс программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем [1].
- Системы контроля версий — это программные инструменты, помогающие командам разработчиков управлять изменениями в исходном коде с течением времени [2].

# Выполнение лабораторной работы

1. Создадим виртуальную машину (fig. 1).

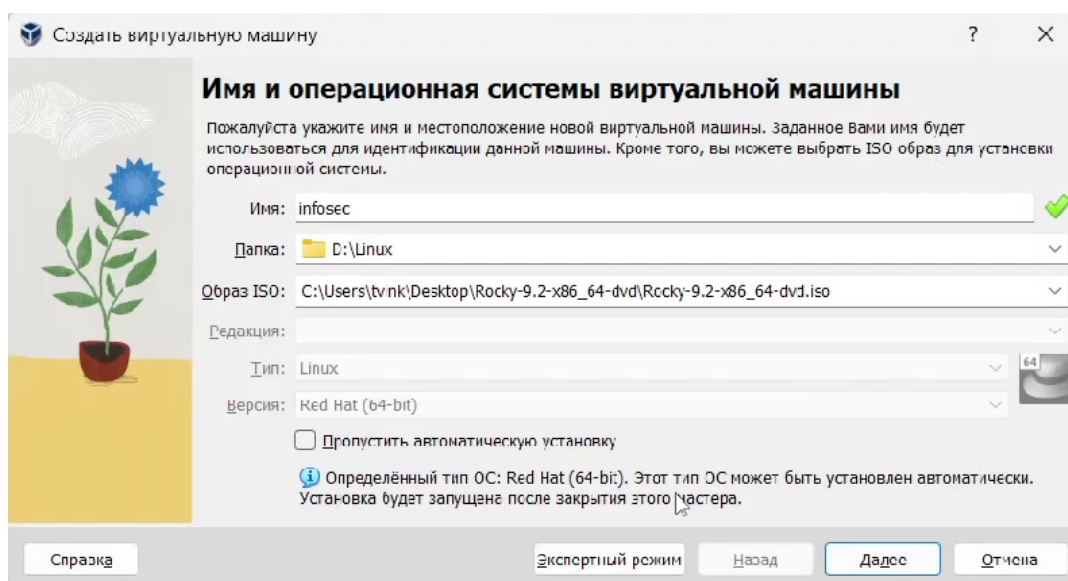


Рис. 1: Окно создания виртуальной машины

2. Запустим виртуальную машину и дождемся загрузки ОС. После этого проведем первичную настройку системы (fig. 2).

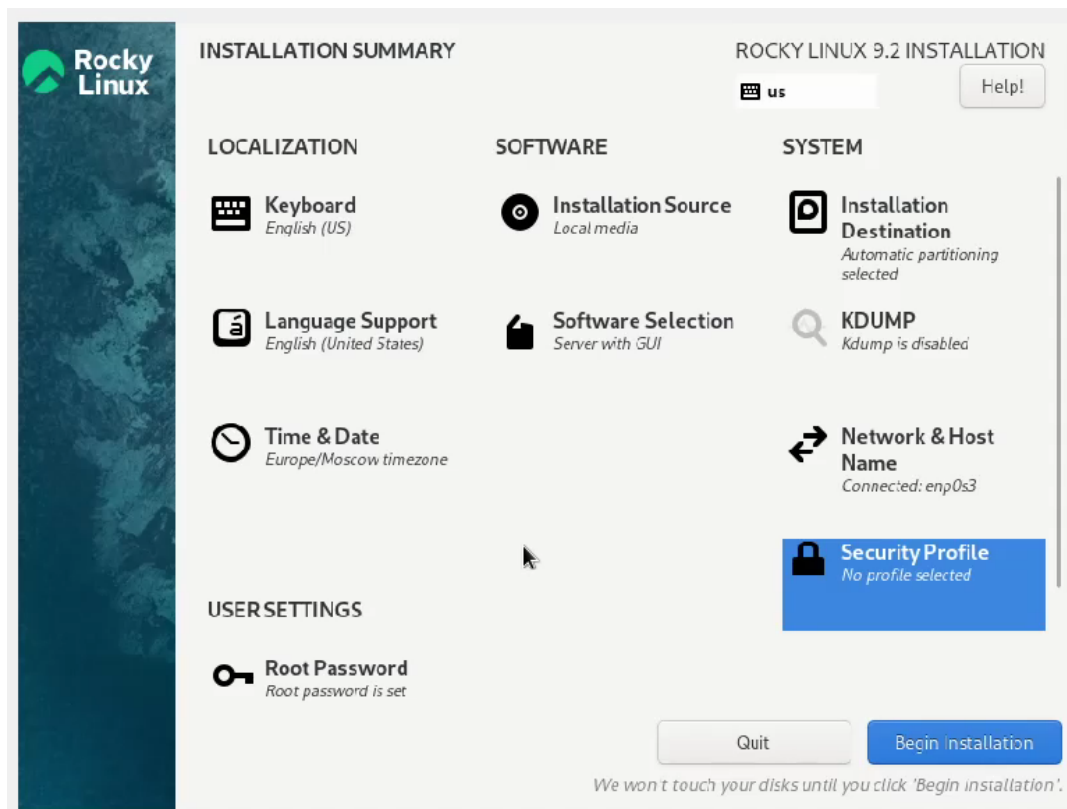


Рис. 2: Окно настройки ОС

- После запуска ОС откроем терминал и выведем некоторую информацию о нашей конфигурации (fig. 3 - fig. 9).

```
[astarusov@user ~]$ dmesg | grep -i "linux version"
[    0.000000] Linux version 5.14.0-284.11.1.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 Tue May 9 17:09:15 UTC 2023
```

Рис. 3: Версия ядра Linux

```
[astarusov@user ~]$ dmesg | grep -i "Mhz processor"
[    0.000009] tsc: Detected 2095.994 MHz processor
[astarusov@user ~]$ dmesg | grep -i "CPU0"
```

Рис. 4: Частота процессора

```
[astarusov@user ~]$ dmesg | grep -i "CPU0"
[    0.237131] smpboot: CPU0: AMD Ryzen 5 5500U with Radeon Graphics (family: 0x17, model: 0x68, stepping: 0x1)
```

Рис. 5: Модель процессора



```

[ 0.002406] ACPI: Reserving SSDT table memory at [mem 0x7fff02a0-0x7fff060b]
[ 0.003974] Early memory node ranges
[ 0.007669] PM: hibernation: Registered nosave memory: [mem 0x00000000-0x00000fff]
[ 0.007672] PM: hibernation: Registered nosave memory: [mem 0x00009f000-0x00009ffff]
[ 0.007673] PM: hibernation: Registered nosave memory: [mem 0x0000a0000-0x0000affff]
[ 0.007673] PM: hibernation: Registered nosave memory: [mem 0x0000b0000-0x0000bffff]
[ 0.020266] Memory: 260866K/209669K available (14342K kernel code, 5536K rwd
ata, 10180K rodata, 2792K init, 7524K bss, 143204K reserved, 0K cma-reserved)
[ 0.132502] Freeing SMP alternatives memory: 36K
[ 0.250282] x86/mm: Memory block size: 128MB
[ 0.426437] Non-volatile memory driver v1.3
[ 0.904294] Freeing initrd memory: 56828K
[ 1.214180] Freeing unused decrypted memory: 2036K
[ 1.214939] Freeing unused kernel image (initmem) memory: 2792K
[ 1.220752] Freeing unused kernel image (text/rodata gap) memory: 2040K
[ 1.221694] Freeing unused kernel image (rodata/data gap) memory: 50K
[ 2.452165] vmwgfx 0000:00:02.0: [drm] Legacy memory limits: VRAM = 16384 kB,
FIFO = 2048 kB, surface = 567904 kB
[ 2.452171] vmwgfx 0000:00:02.0: [drm] Maximum display memory size is 16384 k
iB

```

Рис. 6: Объем доступной оперативной памяти

```

[astarusov@user ~]$ dmesg | grep -i "Hypervisor detected"
[ 0.000000] Hypervisor detected: KVM

```

Рис. 7: Тип обнаруженного гипервизора

```

[astarusov@user ~]$ dmesg | grep -i "Filesystem"
[ 4.753772] XFS (dm-0): Mounting V5 Filesystem
[ 8.879961] XFS (sda1): Mounting V5 Filesystem

```

Рис. 8: Тип файловой системы корневого раздела

```
[astarusov@user ~]$ dmesg | grep -i "mount"
[ 0.123303] Mount-cache hash table entries: 4096 (order: 3, 32768 bytes, linear)
[ 0.123308] Mountpoint-cache hash table entries: 4096 (order: 3, 32768 bytes, linear)
[ 4.753772] XFS (dm-0): Mounting V5 Filesystem
[ 4.799697] XFS (dm-0): Ending clean mount
[ 6.365885] systemd[1]: Set up automount Arbitrary Executable File Formats File System Automount Point.
[ 6.385717] systemd[1]: Mounting Huge Pages File System...
[ 6.387513] systemd[1]: Mounting POSIX Message Queue File System...
[ 6.388445] systemd[1]: Mounting Kernel Debug File System...
[ 6.392576] systemd[1]: Mounting Kernel Trace File System...
[ 6.475638] systemd[1]: Starting Remount Root and Kernel File Systems...
[ 6.513150] systemd[1]: Mounted Huge Pages File System.
[ 6.513580] systemd[1]: Mounted POSIX Message Queue File System.
[ 6.513736] systemd[1]: Mounted Kernel Debug File System.
[ 6.513857] systemd[1]: Mounted Kernel Trace File System.
[ 8.879961] XFS (sda1): Mounting V5 Filesystem
[ 9.210790] XFS (sda1): Ending clean mount
```

Рис. 9: Последовательность монтирования файловых систем

4. Создадим рабочую директорию (fig. 10).

```
[astarusov@user ~]$ mkdir -p ~/work/sLydy/2023-2024/"Информационная безопасность"
```

Рис. 10: Создание рабочей директории

5. Проведем базовую настройку git (fig. 11).

```
[astarusov@user ~]$ git config --global user.name Tvinkvinter
[astarusov@user ~]$ git config --global user.email Tvinkvinter@gmail.com
[astarusov@user ~]$ git config --global core.quotepath false
[astarusov@user ~]$ git config --global init.defaultBranch main
[astarusov@user ~]$ git config --global core.autocrlf input
[astarusov@user ~]$ git config --global core.safecrlf warn
```

Рис. 11: Базовая настройка git

6. Создадим два SSH-ключи по двум разным алгоритмам (fig. 12 - fig. 13).

```
[astarusov@user ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/astarusov/.ssh/id_rsa):
Created directory '/home/astarusov/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/astarusov/.ssh/id_rsa
Your public key has been saved in /home/astarusov/.ssh/id_rsa.pub
```

Рис. 12: Создание ключа SSH по алгоритму rsa

```
[astarusov@user ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/astarusov/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/astarusov/.ssh/id_ed25519
Your public key has been saved in /home/astarusov/.ssh/id_ed25519.pub
```

Рис. 13: Создание ключа SSH по алгоритму ed25519

7. Создадим ключ GPG и добавим его в GitHub (fig. 14 - fig. 16).

```
[astarusov@user ~]$ 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.

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) ☐
```

Рис. 14: Создание ключа GPG

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

Real name:
Email address:
Comment:
You selected this USER-ID:
    ""

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? N
Real name: Tvinkvinter
You selected this USER-ID:
    "Tvinkvinter"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? E
Email address: Tvinkvinter@gmail.com
You selected this USER-ID:
    "Tvinkvinter <Tvinkvinter@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?
```

Рис. 15: Создание ключа GPG

## Add new GPG key

Title

new gpg

Key

```
q709urwE107GAq9Hm1qnpzsoz57ZK0E2C7Dw0p1D11/StatGp0B+ws9s  
Ou39NSVy  
  
cDc6  
  
=xrX9  
  
-----END PGP PUBLIC KEY BLOCK-----
```

Add GPG key

Рис. 16: Добавление ключа GPG в Github

8. Настроим автоматические подписи коммитов git (fig. 17).

```
[astarusov@user ~]$ git config --global user.signingkey 774FD7C4D03A4405  
[astarusov@user ~]$ git config --global commit.gpgsign true  
[astarusov@user ~]$ ^[[200~git config --global gpg.program $(which gpg2)  
bash: git: command not found...  
git config --global gpg.program $(which gpg2)[astarusov@user ~]$ ~  
bash: /home/astarusov: Is a directory  
[astarusov@user ~]$ git config --global gpg.program $(which gpg2)
```

Рис. 17: Настройка автоматических подписей коммитов git

9. Авторизуемся на Github с помощью gh (fig. 18).

```
[astarusov@user ~]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? 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', 'workflow'.
? Paste your authentication token: 
- gh config set -h github.com git_protocol https
/ Configured git protocol
/ Logged in as Tvinkvinter
```

Рис. 18: Настройка gh

10. Создадим репозиторий на основе данного шаблона и склонируем его на локальную машину (fig. 19 - fig. 20).

```
[astarusov@user ~]$ gh repo create study_2023-2024_infosec --template=yamadharma/course-directory-student-template --public
```

Рис. 19: Создание репозитория

```
[astarusov@user Информационная безопасность]$ git clone --recursive git@github.com:tvinkvinter/study_2023-2024_infosec.git infosec
```

Рис. 20: Клонирование репозитория

11. Настроим каталог курса и создадим коммит. Затем отправим изменения на сервер (fig. 21).

```
[astarusov@user infosec]$ rm package.json
[astarusov@user infosec]$ make COURSE=infosec
[astarusov@user infosec]$ git add .
[astarusov@user infosec]$ git commit -am 'feat(main): make course structure'
```

Рис. 21: Настройка каталога курса

## **Выводы**

Были приобретены практические навыки установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов, а также изучены средства контроля версий и получены навыки работы с git.

## **Список литературы**

[1] <https://blog.skillfactory.ru/glossary/operaczionnaya-sistema/>

[2] <https://www.atlassian.com/ru/git/tutorials/what-is-version-control>