

# Отчёт по лабораторной работе №1

## Информационная безопасность

### Настройка рабочего пространства и конфигурация операционной системы на виртуальную машину. Система контроля версий Git. Язык разметки Markdown

Выполнил: Прасолов Валерий,  
НПИбд-02-21, 1032212968

#### Содержание

1	Цель работы.....	1
2	Теоретическое введение.....	2
3	Выполнение лабораторной работы.....	3
3.1	Установка и конфигурация операционной системы на виртуальную машину .....	3
3.1.1	Virtual Box.....	3
3.1.2	Переход в ОС Linux.....	9
3.1.3	Домашнее задание.....	10
3.2	Управление версиями.....	11
3.2.1	Настройка github.....	11
3.2.2	Установка программного обеспечения .....	12
3.2.3	Базовая настройка git .....	12
3.2.4	Создание ключа SSH .....	13
3.2.5	Создание ключа PGP .....	14
3.2.6	Добавление PGP ключа в GitHub .....	14
3.2.7	Настройка автоматических подписей коммитов git.....	15
3.2.8	Настройка gh.....	15
3.2.9	Создание репозитория курса на основе шаблона.....	16
4	Вывод .....	17
5	Список литературы. Библиография.....	18

## 1 Цель работы

Настроить рабочее пространство для лабораторных работ, приобрести практические навыки установки операционной системы на виртуальную машину и настройки

минимально необходимых для дальнейшей работы сервисов. Изучить идеологию и применение средств контроля версий, освоить умения по работе с git. Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

## 2 Теоретическое введение

**Oracle VM VirtualBox** — это мощная и бесплатная виртуализационная платформа, разработанная корпорацией Oracle, которая позволяет пользователям создавать и управлять виртуальными машинами на своих компьютерах. [1]

**Системы контроля версий (Version Control System, VCS)** применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется. [2]

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.

Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом.

Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить.

В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.

Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

Примеры команд для Git:

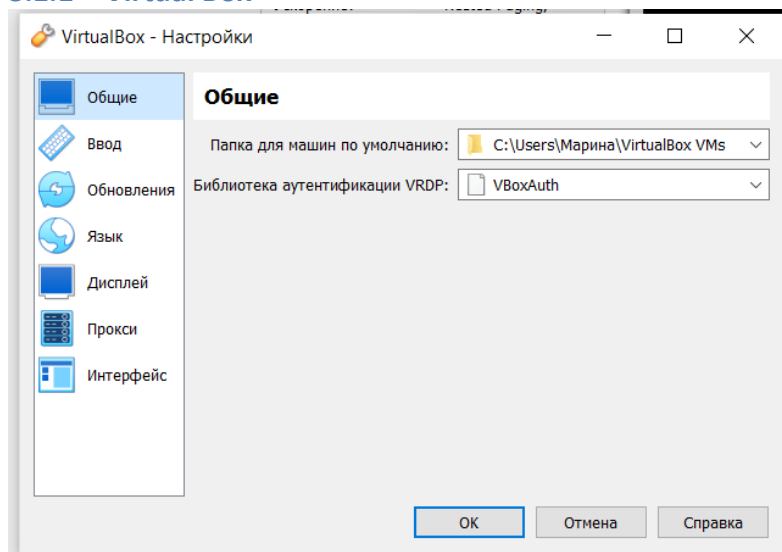
Название команды	Описание команды
<code>git clone</code>	Клонирование репозитория на ПК
<code>git commit -m "Initial Commit"</code>	Оставление коммита
<code>git push</code>	Загрузка изменений на гит
<code>make</code>	Конвертация файла .md

**Markdown** - это легковесный язык разметки, который широко используется для создания форматированного текста в веб-среде. Его простота и читаемость делают его популярным среди разработчиков, писателей и блогеров. Синтаксис Markdown состоит из простых символов и правил форматирования, которые позволяют создавать заголовки, списки, ссылки, изображения и другие элементы веб-страниц без необходимости использовать сложные HTML-теги. Он также легко читается в исходном виде и может быть конвертирован в различные форматы, такие как HTML, PDF или документы Microsoft Word, делая Markdown удобным инструментом для создания содержательного и красочного контента в интернете. [3]

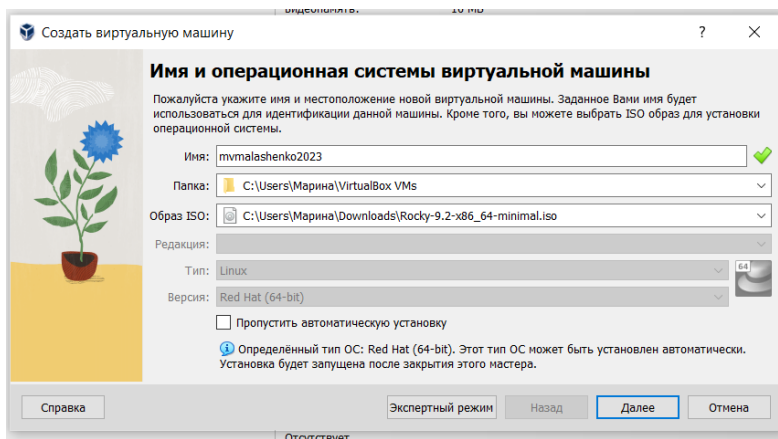
## 3 Выполнение лабораторной работы

### 3.1 Установка и конфигурация операционной системы на виртуальную машину

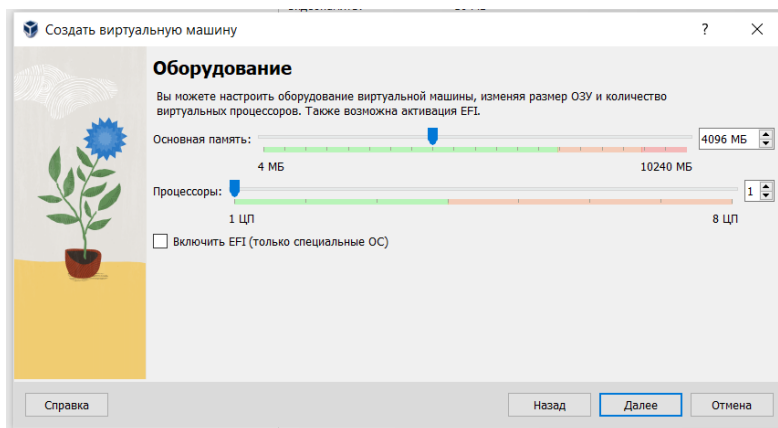
#### 3.1.1 Virtual Box



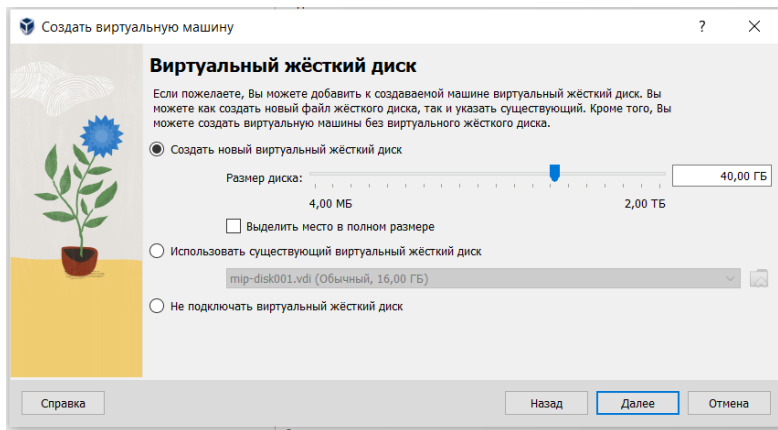
(рис. 1. Общие настройки)



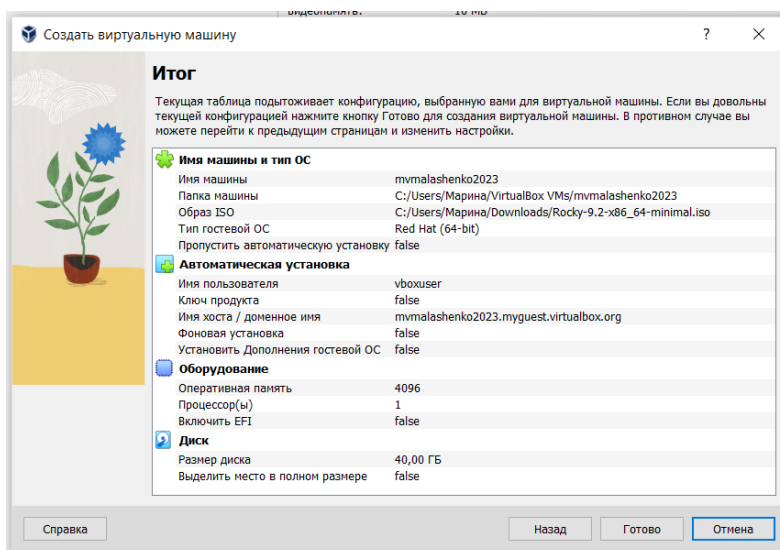
(рис. 2. Имя и путь ОС)



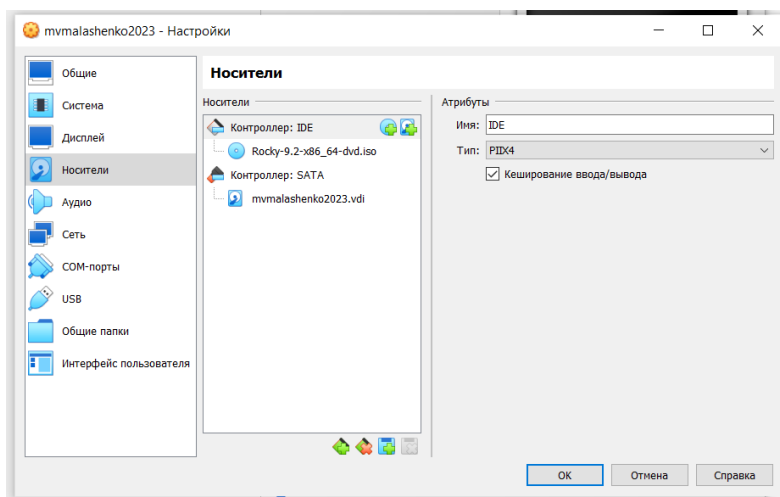
(рис. 3. Размер памяти и число процессоров)



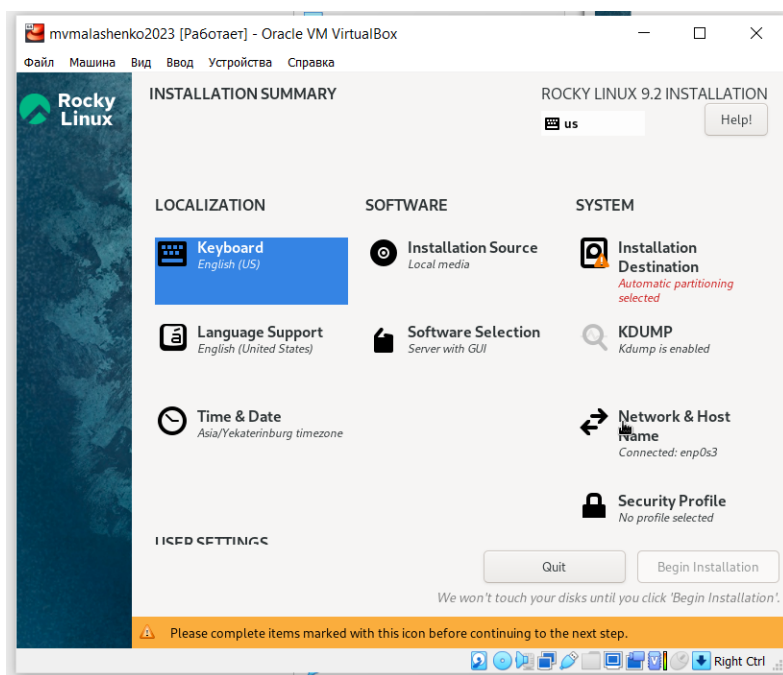
(рис. 4. Виртуальный жесткий диск)



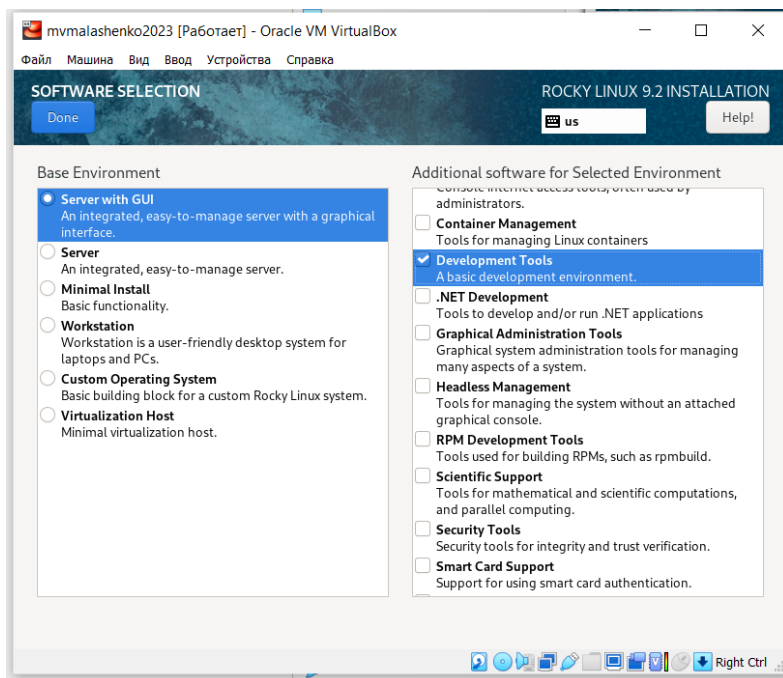
(рис. 5. Итог настроек)



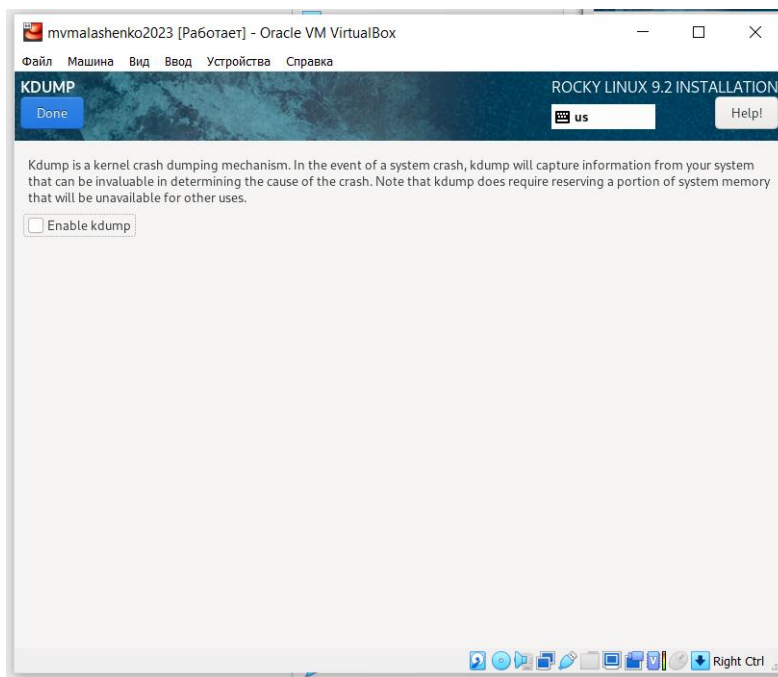
(рис. 6. Носители)



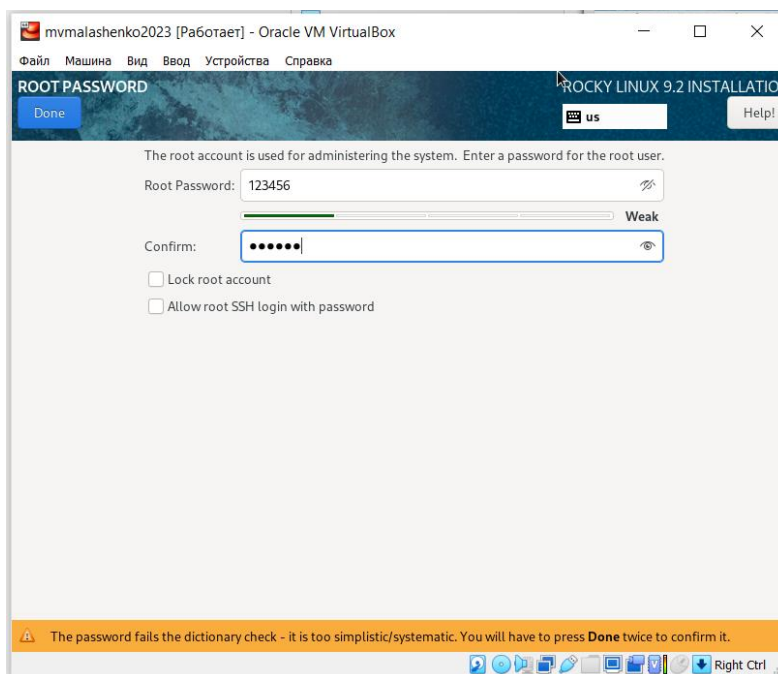
(рис. 7. Стартовое меню установки)



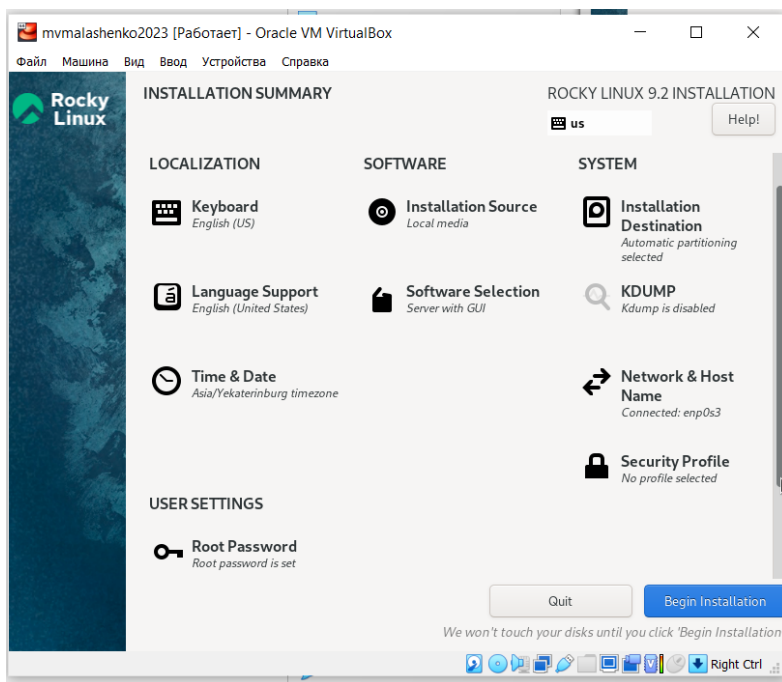
(рис. 8. Server with GUI)



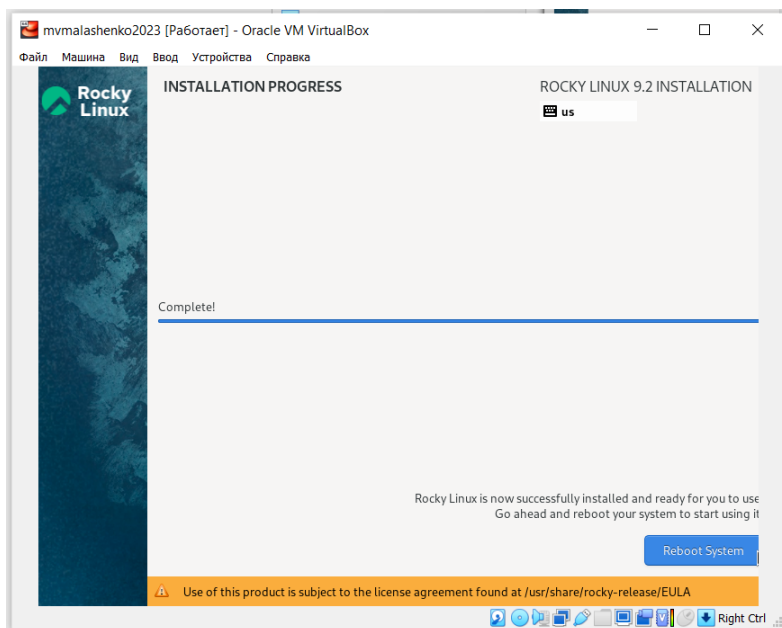
*(puc. 9. Enable KDUMP)*



*(puc. 10. Root password)*



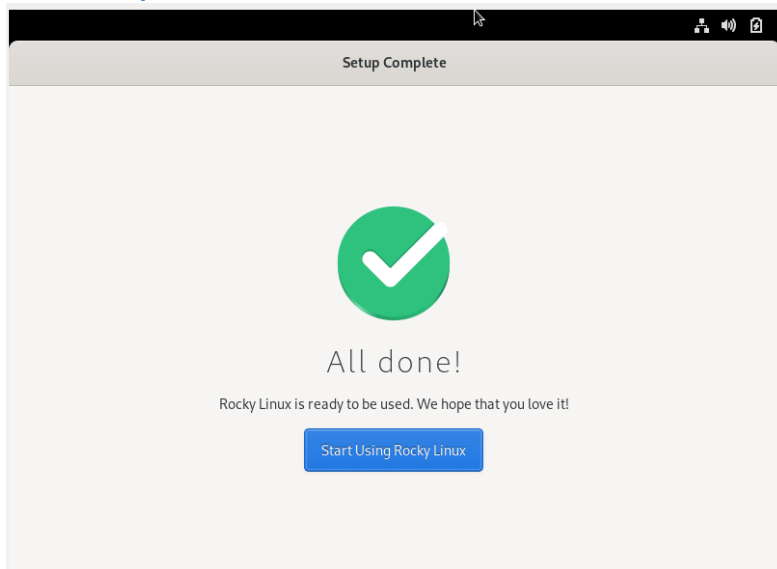
(рис. 11. Итоговое меню установки)



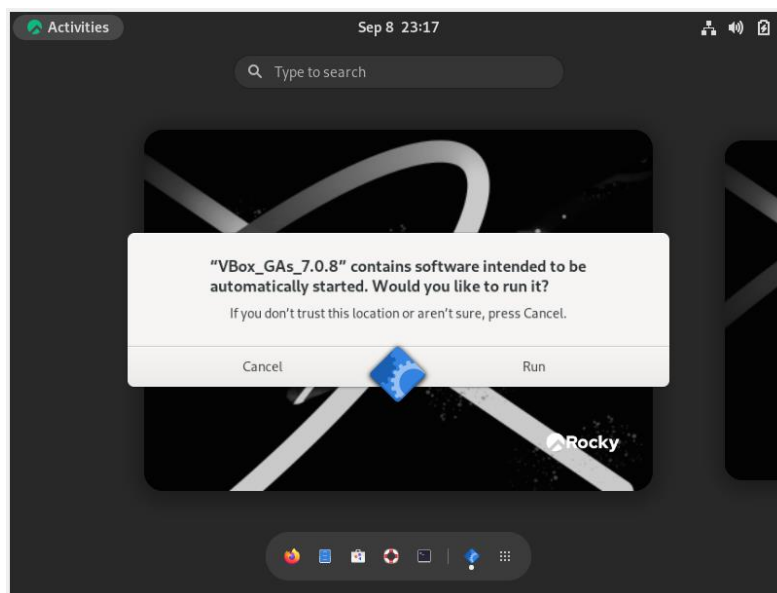
(рис. 12. Успешное завершение установки и перезагрузка системы)



### 3.1.2 Переход в ОС Linux



*(рис. 13. успешное создание пользователя)*



*(рис. 14. Подключение гостевых настроек)*

### 3.1.3 Домашнее задание

```
Activities Terminal Sep 9 00:00
mvmalashenko@mvmalashenko:~
[mvmalashenko@mvmalashenko ~]$ dmesg
[ 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
[ 0.000000] The list of certified hardware and cloud instances for Enterpri
se Linux 9 can be viewed at the Red Hat Ecosystem Catalog, https://catalog.red
hat.com.
[ 0.000000] Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-5.14.0-284.11.1.e
l9_2.x86_64 root=/dev/mapper/rl_mvmalashenko-root ro resume=/dev/mapper/rl_mvm
alashenko-swap rd.lvm.lv=rl_mvmalashenko/root rd.lvm.lv=rl_mvmalashenko/swap r
hgb quiet
[ 0.000000] [Firmware Bug]: TSC doesn't count with P0 frequency!
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point re
gisters'
[ 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 0x00000000000f0000-0x00000000000fffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000100000-0x0000000000dfffff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000dfff0000-0x0000000000dfffffff] ACPI dat
a
[ 0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000ffff0000-0x00000000ffffffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000010000000-0x0000000011fffff] 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/
```

(рис. 15. dmesg)

```
[mvmalashenko@mvmalashenko ~]$ dmesg | less
[1]+ Stopped dmesg | less
[mvmalashenko@mvmalashenko ~]$ 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
[mvmalashenko@mvmalashenko ~]$ dmesg | grep -i "processor"
[ 0.000018] tsc: Detected 2096.064 MHz processor
[ 0.253354] smpboot: Total of 6 processors activated (25152.76 BogoMIPS)
[ 0.267363] ACPI: Added _OSI(Processor Device)
[ 0.267368] ACPI: Added _OSI(Processor Aggregator Device)
[mvmalashenko@mvmalashenko ~]$ dmesg | grep -i "CPU0"
[ 0.232235] smpboot: CPU0: AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx (f
amily: 0x17, model: 0x18, stepping: 0x1)
[mvmalashenko@mvmalashenko ~]$
```

(рис. 16. dmesg | less, версия ядра линукс, частота процессора, модель процессора)

```
[mvmalashenko@mvmalashenko ~]$ dmesg | grep -i "memory"
[ 0.002007] ACPI: Reserving FACP table memory at [mem 0xdfff00f0-0xdfff01e3]
[ 0.002008] ACPI: Reserving DSDT table memory at [mem 0xdfff0630-0xdfff2982]
[ 0.002009] ACPI: Reserving FACS table memory at [mem 0xdfff0200-0xdfff023f]
[ 0.002010] ACPI: Reserving FACS table memory at [mem 0xdfff0200-0xdfff023f]
[ 0.002011] ACPI: Reserving APIC table memory at [mem 0xdfff0240-0xdfff02bb]
[ 0.002012] ACPI: Reserving SSDT table memory at [mem 0xdfff02c0-0xdfff062b]
[ 0.002509] Early memory node ranges
[ 0.029553] PM: hibernation: Registered nosave memory: [mem 0x00000000-0x00000fff]
[ 0.029557] PM: hibernation: Registered nosave memory: [mem 0x0009f000-0x0009ffff]
[ 0.029558] PM: hibernation: Registered nosave memory: [mem 0x000a0000-0x000effff]
[ 0.029558] PM: hibernation: Registered nosave memory: [mem 0x000f0000-0x000fffff]
[ 0.029559] PM: hibernation: Registered nosave memory: [mem 0xdfff0000-0xdffffffffff]
[ 0.029560] PM: hibernation: Registered nosave memory: [mem 0xe0000000-0xfefbffff]
[ 0.029561] PM: hibernation: Registered nosave memory: [mem 0xfec00000-0xfefc00fff]
```

(рис. 17. Объем доступной оперативной памяти)

```
[mvmalashenko@mvmalashenko ~]$ dmesg | grep -i "hypervisor"
[ 0.000000] Hypervisor detected: KVM
```

(рис. 18. Тип обнаруженного гипервизора)

```
[mvmalashenko@mvmalashenko ~]$ dmesg | grep -i "VFS: Mounted root"
[mvmalashenko@mvmalashenko ~]$ df -T
```

Filesystem	Type	1K-blocks	Used	Available	Use%	Mounted on
devtmpfs	devtmpfs	4096	0	4096	0%	/dev
tmpfs	tmpfs	2005696	0	2005696	0%	/dev/shm
tmpfs	tmpfs	802280	10112	792168	2%	/run
/dev/mapper/rl_mvmalashenko-root	xfs	36727276	5447636	31279640	15%	/
/dev/sda1	xfs	1038336	263524	774812	26%	/boot
tmpfs	tmpfs	401136	120	401016	1%	/run
/user/1000						
/dev/sr0	iso9660	52232	52232	0	100%	/run
media/mvmalashenko/VBox_GAs_7.0.8						

(рис. 19. Тип файловой системы корневого раздела)

```
[mvmalashenko@mvmalashenko ~]$ dmesg | grep -i "Mounted"
[ 7.058615] systemd[1]: Mounted Huge Pages File System.
[ 7.059273] systemd[1]: Mounted POSIX Message Queue File System.
[ 7.059873] systemd[1]: Mounted Kernel Debug File System.
[ 7.060609] systemd[1]: Mounted Kernel Trace File System.
[mvmalashenko@mvmalashenko ~]$ mount
```

(рис. 20. Последовательность монтирования файловых систем)

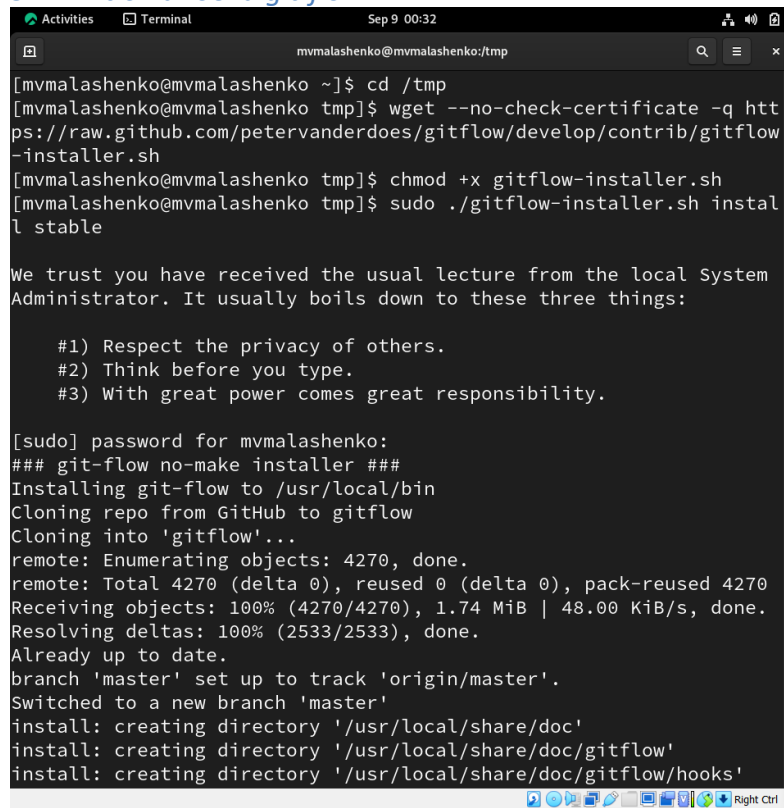
## 3.2 Управление версиями

### 3.2.1 Настройка github

Уже имеется аккаунт: <https://github.com/Malashenkov>

## 3.2.2 Установка программного обеспечения

### 3.2.2.1 Установка git-flow



```
[mvmalashenko@mvmalashenko ~]$ cd /tmp
[mvmalashenko@mvmalashenko tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
[mvmalashenko@mvmalashenko tmp]$ chmod +x gitflow-installer.sh
[mvmalashenko@mvmalashenko tmp]$ sudo ./gitflow-installer.sh install stable

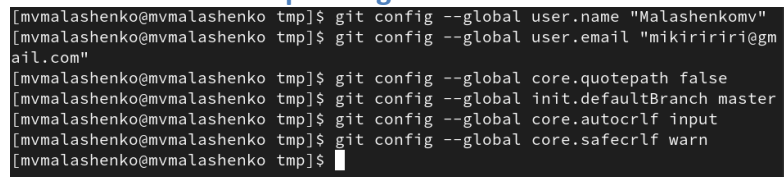
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for mvmalashenko:
### git-flow no-make installer ###
Installing git-flow to /usr/local/bin
Cloning repo from GitHub to gitflow
Cloning into 'gitflow'...
remote: Enumerating objects: 4270, done.
remote: Total 4270 (delta 0), reused 0 (delta 0), pack-reused 4270
Receiving objects: 100% (4270/4270), 1.74 MiB | 48.00 KiB/s, done.
Resolving deltas: 100% (2533/2533), done.
Already up to date.
branch 'master' set up to track 'origin/master'.
Switched to a new branch 'master'
install: creating directory '/usr/local/share/doc'
install: creating directory '/usr/local/share/doc/gitflow'
install: creating directory '/usr/local/share/doc/gitflow/hooks'
```

(рис. 21. Установка git-flow)

### 3.2.3 Базовая настройка git



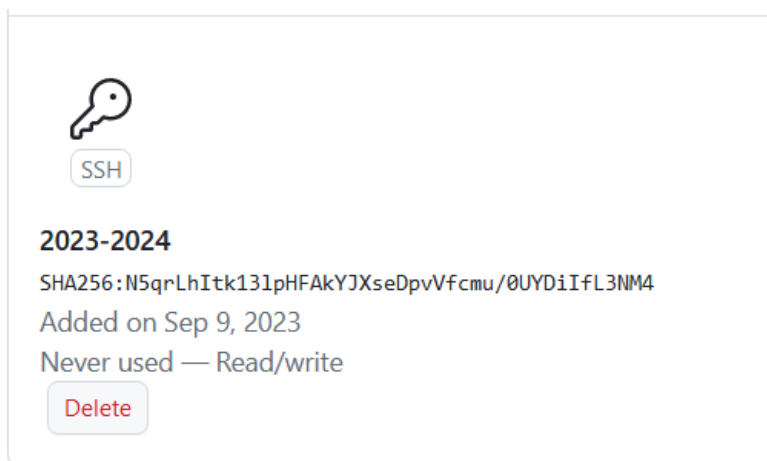
```
[mvmalashenko@mvmalashenko tmp]$ git config --global user.name "Malashenko mv"
[mvmalashenko@mvmalashenko tmp]$ git config --global user.email "mikiririri@gmail.com"
[mvmalashenko@mvmalashenko tmp]$ git config --global core.quotepath false
[mvmalashenko@mvmalashenko tmp]$ git config --global init.defaultBranch master
[mvmalashenko@mvmalashenko tmp]$ git config --global core.autocrlf input
[mvmalashenko@mvmalashenko tmp]$ git config --global core.safecrlf warn
[mvmalashenko@mvmalashenko tmp]$
```

(рис. 22. Базовая настройка git)

### 3.2.4 Создание ключа SSH

```
[mvmalashenko@mvmalashenko ~]$ ssh-keygen -t rsa -b 4096 -C "mikiririri@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/mvmalashenko/.ssh/id_rsa):
/home/mvmalashenko/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/mvmalashenko/.ssh/id_rsa
Your public key has been saved in /home/mvmalashenko/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:N5qrLhItk13lpHFAkYJXseDpvVfcmu/0UYDiIfL3NM4 mikiririri@gmail.com
The key's randomart image is:
+---[RSA 4096]-----+
|  .O*=                |
| ..000.+ .            |
| .o.oB. o . .         |
|  .oo.+ + .           |
| +... S B + .         |
| = o . * O . .        |
| + . + o E .          |
| . . . . o . .        |
| . oo.. .o .          |
+-----[SHA256]-----+
```

(рис. 23. Создание ключа SSH)



(рис. 24. Ключ SSH успешно добавлен на git)

### 3.2.5 Создание ключа PGP

```
[mvmalashenko@mvmalashenko ~]$ 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/mvmalashenko/.gnupg' created
gpg: keybox '/home/mvmalashenko/.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.
```

(рис. 25. Создание ключа PGP(1))

### 3.2.6 Добавление PGP ключа в GitHub

```
Real name: malashenkov
Email address: mikiririri@gmail.com
Comment: 2023-2024
You selected this USER-ID:
    "malashenkov (2023-2024) <mikiririri@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: /home/mvmalashenko/.gnupg/trustdb.gpg: trustdb created
gpg: key 8209748AF8E33F86 marked as ultimately trusted
gpg: directory '/home/mvmalashenko/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/mvmalashenko/.gnupg/openpgp-revoc
s.d/2C117090C864942DCC5A309B8209748AF8E33F86.rev'
public and secret key created and signed.

pub   rsa4096 2023-09-08 [SC]
       2C117090C864942DCC5A309B8209748AF8E33F86
uid           malashenkov (2023-2024) <mikiririri@gmail.com>
sub   rsa4096 2023-09-08 [E]

[mvmalashenko@mvmalashenko ~]$
```

(рис. 26. Создание ключа PGP(2))


```
[mvmalashenko@mvmalashenko ~]$ gpg --armor --export <2C117090C864942DCC5A309B8
209748AF8E33F86> | xclip -sel clip
bash: syntax error near unexpected token `|'
[mvmalashenko@mvmalashenko ~]$ gpg --armor --export 2C117090C864942DCC5A309B82
09748AF8E33F86 | xclip -sel clip
[mvmalashenko@mvmalashenko ~]$
```

(рис. 27. Создание ключа PGP(3))

## GPG keys

[New GPG key](#)

This is a list of GPG keys associated with your account. Remove any keys that you do not recognize.



**2023-2024**

Email address: mikiririri@gmail.com

Key ID: 8209748AF8E33F86

Subkeys: FA3E9CB2B772FEF6

Added on Sep 9, 2023

Delete

Learn how to [generate a GPG key and add it to your account](#).

(рис. 28. Ключ PGP успешно добавлен на git)

### 3.2.7 Настройка автоматических подписей коммитов git

```
[mvmalashenko@mvmalashenko ~]$ git config --global user.signingkey 2C117090C864942DCC5A309B8209748AF8E33F86
[mvmalashenko@mvmalashenko ~]$ git config --global commit.gpgsign true
[mvmalashenko@mvmalashenko ~]$ git config --global gpg.program $(which gpg2)
[mvmalashenko@mvmalashenko ~]$
```

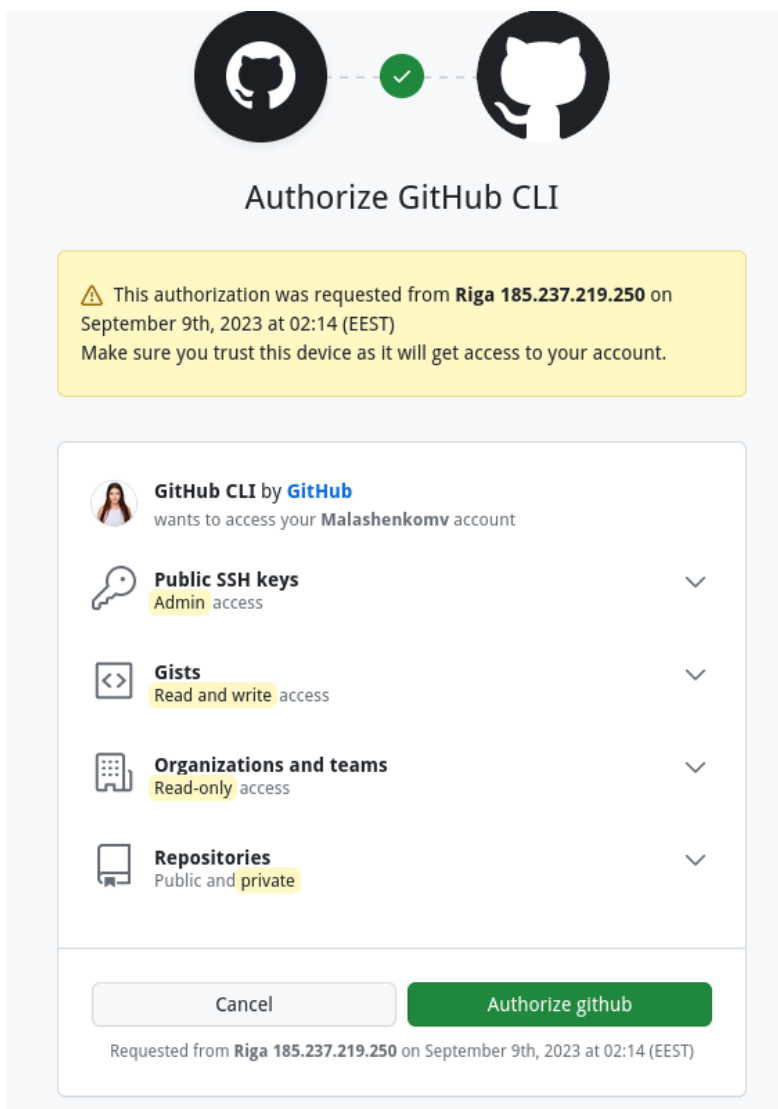
(рис. 29. Настройка авто-коммитов)

### 3.2.8 Настройка gh

```
[mvmalashenko@mvmalashenko ~]$ gh auth login
? What account do you want to log into? GitHub.com
? You're already logged into github.com. Do you want to re-authenticate? No
[mvmalashenko@mvmalashenko ~]$ gh auth login
? What account do you want to log into? GitHub.com
? You're already logged into github.com. Do you want to re-authenticate? Yes
? What is your preferred protocol for Git operations? SSH
? Upload your SSH public key to your GitHub account? /home/mvmalashenko/.ssh/id_rsa.pub
? Title for your SSH key: 2023-2024
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: A8F3-C985
Press Enter to open github.com in your browser...
Missing chrome or resource URL: resource://gre/modules/UpdateListener.jsm
Missing chrome or resource URL: resource://gre/modules/UpdateListener.sys.mjs
✓ Authentication complete.
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ SSH key already existed on your GitHub account: /home/mvmalashenko/.ssh/id_rsa.pub
✓ Logged in as MalashenkoMv
[mvmalashenko@mvmalashenko ~]$
```

(рис. 30. Авторизация gh)



(рис. 31. Успешная авторизация gh)

### 3.2.9 Создание репозитория курса на основе шаблона

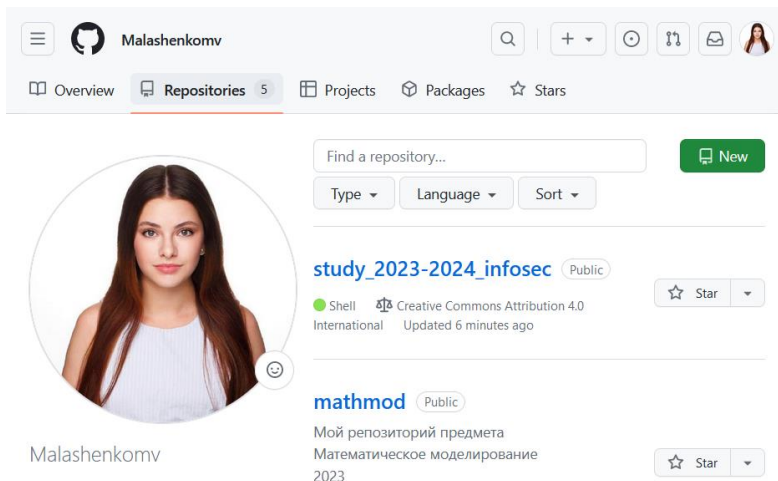
```
[mvmalashenko@mvmalashenko Информационная безопасность]$ gh repo create study_2023-2024_infosec --template=yamadharma/course-directory-student-template --public
✓ Created repository Malashenkov/study_2023-2024_infosec on GitHub
```

(рис. 32. Создание репозитория по шаблону)



```
[mvmalashenko@mvmalashenko Информационная безопасность]$ git clone --recursive
git@github.com:malashenkov/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:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 27 (delta 1), reused 11 (delta 0), pack-reused 0
Receiving objects: 100% (27/27), 16.93 KiB | 2.42 MiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharma/academic-pres
entation-markdown-template.git) registered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharma/academic-laboratory
-report-template.git) registered for path 'template/report'
Cloning into '/home/mvmalashenko/work/2023-2024/Информационная безопасность/in
fosec/template/presentation'...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 82 (delta 28), reused 77 (delta 23), pack-reused 0
Receiving objects: 100% (82/82), 92.90 KiB | 191.00 KiB/s, done.
Resolving deltas: 100% (28/28), done.
Cloning into '/home/mvmalashenko/work/2023-2024/Информационная безопасность/in
fosec/template/report'...
remote: Enumerating objects: 101, done.
remote: Counting objects: 100% (101/101), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 101 (delta 40), reused 88 (delta 27), pack-reused 0
Receiving objects: 100% (101/101), 327.25 KiB | 265.00 KiB/s, done.
Resolving deltas: 100% (40/40), done.
Submodule path 'template/presentation': checked out 'b1be3800ee91f5809264cb755'
```

(рис. 33. Рекурсивное клонирование репозитория)



(рис. 34. Репозиторий успешно создан)

Далее дополнительно настроили репозиторий курса согласно заданию, написали отчет на Markdown и конвертировали .md в .pdf и .docs при помощи команды make и файла Makefile.

Отправили все изменения и коммиты на GitHub.

## 4 Вывод

Были настроено рабочее пространство для лабораторных работ, приобретены практические навыки установки операционной системы на виртуальную машину и настройки минимально необходимых для дальнейшей работы сервисов. Были изучены

идеология и применение средств контроля версий, освоены умения по работе с git. Были приобретены практические навыки оформления отчётов с помощью легковесного языка разметки Markdown.

## 5 Список литературы. Библиография

[1] Документация по Virtual Box: <https://www.virtualbox.org/wiki/Documentation>

[2] Документация по Git: <https://git-scm.com/book/ru/v2>

[3] Документация по Markdown: <https://learn.microsoft.com/ru-ru/contribute/markdown-reference>