

# Отчет по лабораторной работе №2

## Операционные системы

Гаязов Рузаль Ильшатovich

### Содержание

Цель работы .....	1
Задание.....	1
Выполнение лабораторной работы .....	2
Установка программного обеспечения .....	2
Базовая настройка git.....	2
Создание ключа SSH .....	3
Создание ключа GPG.....	4
Регистрация на Github.....	5
Добавление ключа GPG в Github.....	5
Настроить подписи Git.....	7
Настройка gh .....	7
Создание репозитория курса на основе шаблона.....	8
Выводы .....	10
Ответы на контрольные вопросы. ....	10
Список литературы.....	12

### Цель работы

Цель данной лабораторной работы – изучение идеологии и применения средств контроля версий, освоение умения по работе с git.

### Задание

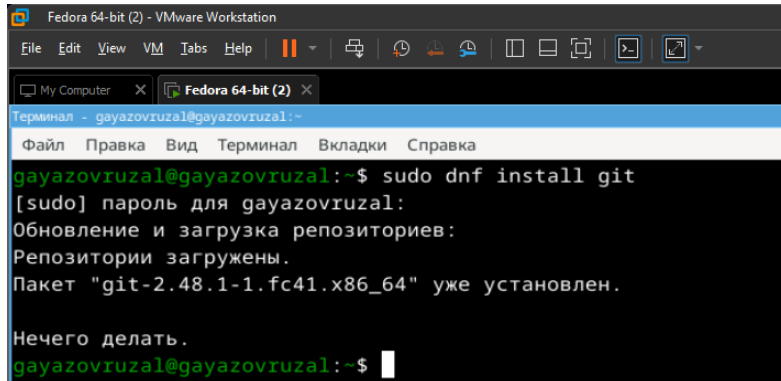
1. Создать базовую конфигурацию для работы с git
2. Создать ключ SSH
3. Создать ключ GPG
4. Настроить подписи Git
5. Зарегистрироваться на GitHub

6. Создать локальный каталог для выполнения заданий по предмету.

## Выполнение лабораторной работы

### Установка программного обеспечения

Устанавливаю необходимое программное обеспечение git и gh через терминал с помощью команд: `dnf install git` и `dnf install gh` (рис. @fig:001).

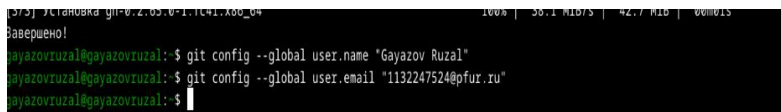


```
Fedora 64-bit (2) - VMware Workstation
File Edit View VM Tabs Help
Терминал - gayazovruzal@gayazovruzal:~
Файл Правка Вид Терминал Вкладки Справка
gayazovruzal@gayazovruzal:~$ sudo dnf install git
[sudo] пароль для gayazovruzal:
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет "git-2.48.1-1.fc41.x86_64" уже установлен.
Нечего делать.
gayazovruzal@gayazovruzal:~$
```

*Установка git и gh*

### Базовая настройка git

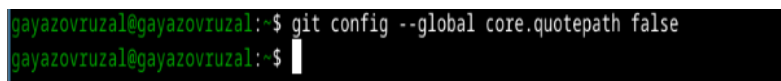
Задаю в качестве имени и email владельца репозитория свои имя, фамилию и электронную почту (рис. @fig:002).



```
[53] Установка git-2.48.1-1.fc41.x86_64 100% | 38.1 MiB/s | 42.7 MiB | 00:01
Завершено!
gayazovruzal@gayazovruzal:~$ git config --global user.name "Gayazov Ruzal"
gayazovruzal@gayazovruzal:~$ git config --global user.email "1132247524@pfur.ru"
gayazovruzal@gayazovruzal:~$
```

*Задаю имя и email владельца репозитория*

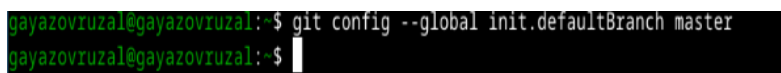
Настраиваю utf-8 в выводе сообщений git для их корректного отображения (рис. @fig:003).



```
gayazovruzal@gayazovruzal:~$ git config --global core.quotepath false
gayazovruzal@gayazovruzal:~$
```

*Настройка utf-8 в выводе сообщений git*

Начальной ветке задаю имя master (рис. @fig:004).



```
gayazovruzal@gayazovruzal:~$ git config --global init.defaultBranch master
gayazovruzal@gayazovruzal:~$
```

*Задаю имя начальной ветки*

Задаю параметры autocrlf и safecrlf (рис. @fig:005).

```
gayazovruzal@gayazovruzal:~$ git config --global core.safecrlf warn
gayazovruzal@gayazovruzal:~$ S
```

*Задаю параметры autocrlf и safecrlf*

## Создание ключа SSH

Создаю ключ ssh размером 4096 бит по алгоритму rsa (рис. @fig:006).

```
gayazovruzal@gayazovruzal:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/gayazovruzal/.ssh/id_rsa):
Created directory '/home/gayazovruzal/.ssh'.
Enter passphrase for "/home/gayazovruzal/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/gayazovruzal/.ssh/id_rsa
Your public key has been saved in /home/gayazovruzal/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:T01y0c7032MFdP4/BSmvFDLWAKrT+ffi85My3DWKQ0o gayazovruzal@gayazovruzal
The key's randomart image is:
+---[RSA 4096]-----+
|  . . . . |
|  o +   .o . |
|  . o o..o+ + |
|  . *== * . |
|  S..=B o. |
|  . + o =.= |
|  . o+.+.+* |
|  E .B.+++ |
|              =o.o |
+---[SHA256]-----+
gayazovruzal@gayazovruzal:~$ S
```

*Генерация ssh ключа по алгоритму rsa*

Создаю ключ ssh по алгоритму ed25519 (рис. @fig:007).

```

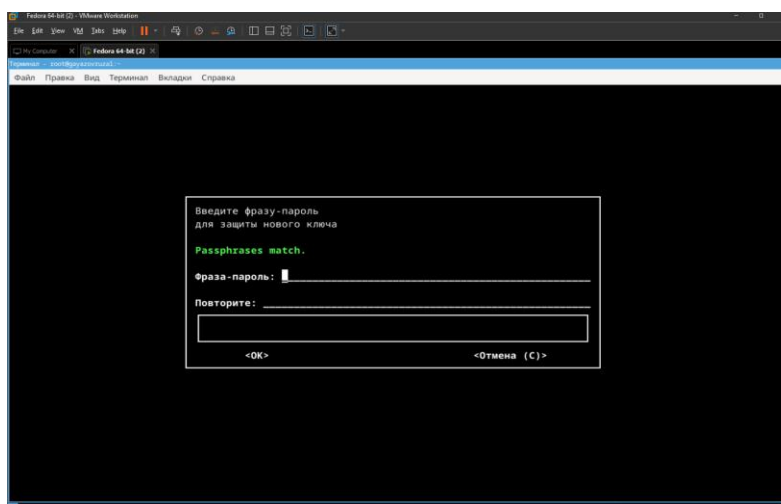
gayazovruzal@gayazovruzal:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/gayazovruzal/.ssh/id_ed25519):
Enter passphrase for "/home/gayazovruzal/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/gayazovruzal/.ssh/id_ed25519
Your public key has been saved in /home/gayazovruzal/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:h490SXo91USpGpBHMxwR2WyfF/ZAWn0ZucwiIihAtc gayazovruzal@gayazovruzal
The key's randomart image is:
+--[ED25519 256]--+
|      .  ++..o*o  |
|      . E.  o 000B. +|
|      . .  o .o.oB0o.o|
|      . . . 00.+@ o.|
|      . S .+=+ * .|
|      .  o *. . .|
|      .  = + .|
|      .  + .|
|      .  .|
+-----[SHA256]-----+
gayazovruzal@gayazovruzal:~$

```

Генерация ssh ключа по алгоритму ed25519

## Создание ключа GPG

Генерирую ключ GPG, затем выбираю тип ключа RSA and RSA, задаю максимальную длину ключа: 4096, оставляю неограниченный срок действия ключа. Далее отвечаю на вопросы программы о личной информации (рис. @fig:008).



Генерация ключа

Ввожу фразу-пароль для защиты нового ключа (рис. @fig:009).

```

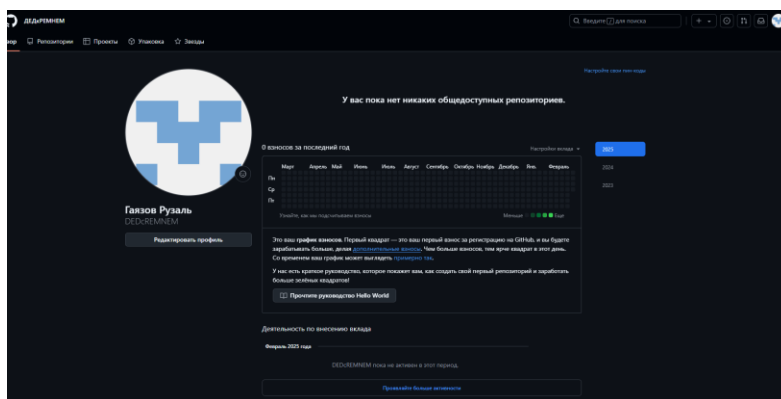
gayazovruzal@gayazovruzal:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/gayazovruzal/.ssh/id_ed25519):
Enter passphrase for "/home/gayazovruzal/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/gayazovruzal/.ssh/id_ed25519
Your public key has been saved in /home/gayazovruzal/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:h490SXo91USpGpBHMxwR2WYfF/ZAWn0ZucwiIihAtc gayazovruzal@gayazovruzal
The key's randomart image is:
+--[ED25519 256]--+
|      ++..0*o |
| . . E. 0 000B. +|
|.. . 0 .0.0B00.0|
| . . . . 00.+@ 0.|
| . . . S .+=+ * .|
|      0 *. . . |
|      . = +    |
|      + .      |
|      .        |
+-----[SHA256]-----+
gayazovruzal@gayazovruzal:~$

```

*Защита ключа GPG*

## Регистрация на Github

У меня уже был создан аккаунт на Github, соответственно, основные данные аккаунта я так же заполняла и проводила его настройку, поэтому просто вхожу в свой аккаунт (рис. @fig:010).



*Аккаунт на Github*

## Добавление ключа GPG в Github

Вывожу список созданных ключей в терминал, ищу в результате запроса отпечаток ключа, он стоит после знака следа, копирую его в буфер обмена (рис. @fig:011).

```

root@gayazovruzal:~# gpg --list-secret-keys --keyid-format LONG
gpg: проверка таблицы доверия
gpg: marginals needed: 3 completes needed: 1 trust model: gpg
gpg: глубина: 0 достоверных: 1 подписанных: 0 доверие: 0-, 0q, 0n, 0m, 0f, 1u
[keyboxd]
-----
sec   rsa4096/0BFA55E0DB530D77 2025-02-26 [SC]
      A2D7D2402D07B9D339D37B3D0BFA55E0DB530D77
uid    [ абсолютно ] gayazovruzal <salinwer@mail.ru>
ssb    rsa4096/72C91C6E1726F78B 2025-02-26 [E]

root@gayazovruzal:~#

```

### Вывод списка ключей

Ввожу в терминале команду, с помощью которой копирую сам ключ GPG в буфер обмена, за это отвечает утилита xclip (рис. @fig:012).

```

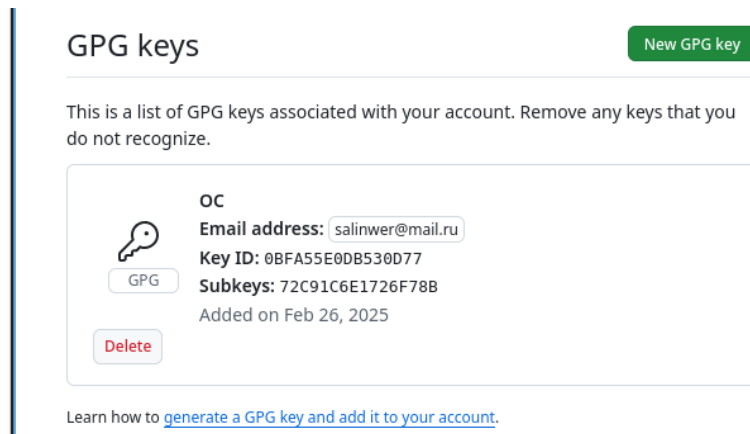
Файл  Правка  Вид  Терминал  Вкладки  Справка
gayazovruzal@gayazovruzal:~$ gayazovruzal@gayazovruzal:
gayazovruzal@gayazovruzal:~$ sudo -i
[sudo] пароль для gayazovruzal:
root@gayazovruzal:~# gpg --list-secret-keys --keyid-format LONG
[keyboxd]
-----
sec   rsa4096/0BFA55E0DB530D77 2025-02-26 [SC]
      A2D7D2402D07B9D339D37B3D0BFA55E0DB530D77
uid    [ абсолютно ] gayazovruzal <salinwer@mail.ru>
ssb    rsa4096/72C91C6E1726F78B 2025-02-26 [E]

root@gayazovruzal:~# gpg --armor --export 0BFA55E0DB530D77 | xclip -sel clip
root@gayazovruzal:~#

```

### Копирование ключа в буфер обмена

Открываю настройки GitHub, ищу среди них добавление GPG ключа. Нажимаю на “New GPG key” и вставляю в поле ключ из буфера обмена (рис. @fig:013).



## Настройка GitHub

### Настроить подписи Git

Настраиваю автоматические подписи коммитов git: используя введенный ранее email, указываю git использовать его при создании подписей коммитов (рис. @fig:016).

```
gayazovruzal@gayazovruzal:~$ git config global user.signingkey 0BFA55E0DB530D77
fatal: not in a git directory
gayazovruzal@gayazovruzal:~$ git config --global user.signingkey 0BFA55E0DB530D77
gayazovruzal@gayazovruzal:~$ git config commit.gpgsign true
fatal: not in a git directory
gayazovruzal@gayazovruzal:~$ git config --global commit.gpgsign true
gayazovruzal@gayazovruzal:~$ git config --global gpg.program $(which gpg2)
gayazovruzal@gayazovruzal:~$ gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? [Use arrows to m
> HTTPS
SSH
```

## Настройка подписей Git

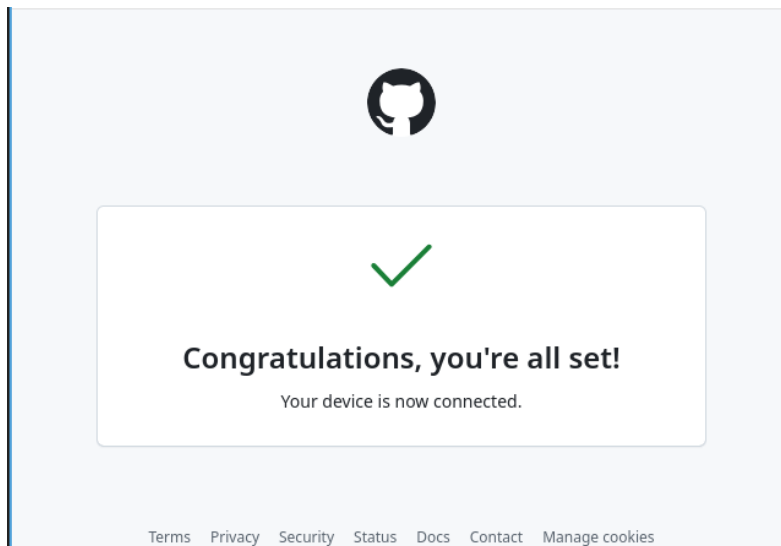
### Настройка gh

Начинаю авторизацию в gh, отвечаю на наводящие вопросы от утилиты, в конце выбираю авторизоваться через браузер (рис. @fig:017).

```
gayazovruzal@gayazovruzal:~$ git config global user.signingkey 0BFA55E0DB530D77
fatal: not in a git directory
gayazovruzal@gayazovruzal:~$ git config --global user.signingkey 0BFA55E0DB530D77
gayazovruzal@gayazovruzal:~$ git config commit.gpgsign true
fatal: not in a git directory
gayazovruzal@gayazovruzal:~$ git config --global commit.gpgsign true
gayazovruzal@gayazovruzal:~$ git config --global gpg.program $(which gpg2)
gayazovruzal@gayazovruzal:~$ gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? [Use arrows to m
> HTTPS
SSH
```

## Авторизация в gh

Завершаю авторизацию на сайте (рис. @fig:018).



### *Завершение авторизации через браузер*

Вижу сообщение о завершении авторизации под именем evdvorkina (рис. @fig:019).

```
gayazovrusal@gayazovrusal:~$ gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser

First copy your one-time code: 62E6-B140
Press Enter to open https://github.com/login/device in your browser...
restorecon: SELinux: Could not get canonical path for /home/gayazovrusal/.mozilla/firefox/* /gmp-widevinecdm/* restorecon: No
such file or directory.
(GFX1-): RenderCompositorSWGL failed mapping default framebuffer, no dt

Authentication complete.
- gh config set -h github.com git_protocol https
- Configured git protocol
- Authentication credentials saved in plain text
Logged in as evdvorkina
gayazovrusal@gayazovrusal:~$
```

### *Завершение авторизации*

## Создание репозитория курса на основе шаблона

Сначала создаю директорию с помощью утилиты `mkdir` и флага `-p`, который позволяет установить каталоги на всем указанном пути. После этого с помощью утилиты `cd` перехожу в только что созданную директорию “Операционные системы”. Далее в терминале ввожу команду `gh repo create study_2022-2023_os-intro --template uamadharm/course-directory-student-trmplate --public`, чтобы создать репозиторий на основе шаблона репозитория. После этого клонирую репозиторий к себе в директорию, я указываю ссылку с протоколом `https`, а не `ssh`, потому что при авторизации в `gh` выбрала протокол `https` (рис. @fig:020).



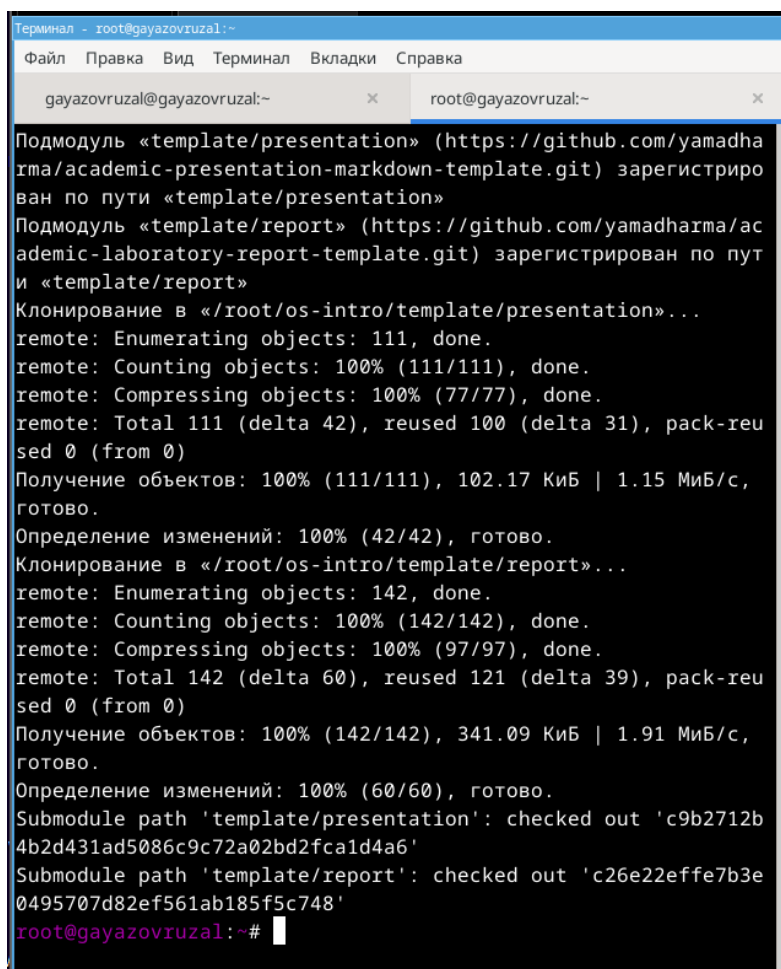
```
gayazovruzal@gayazovruzal:~$ gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 62E5-B140
Press Enter to open https://github.com/login/device in your browser...
restorecon: SELinux: Could not get canonical path for /home/gayazovruzal/.mozilla/firefox/*: restorecon: No
such file or directory.
[GTK-3]: RenderCompositorSWGL failed mapping default framebuffer, no dt

. Authentication complete.
. gh config set -h github.com git_protocol https
. Configured git protocol
. Authentication credentials saved in plain text
. Logged in as DEDcREMNEH
gayazovruzal@gayazovruzal:~$
gayazovruzal@gayazovruzal:~$
```

## Создание репозитория

Перехожу в каталог курса с помощью утилиты `cd`, проверяю содержание каталога с помощью утилиты `ls` (рис. @fig:021).



```
Терминал - root@gayazovruzal:~
Файл  Правка  Вид  Терминал  Вкладки  Справка

gayazovruzal@gayazovruzal:~  x  root@gayazovruzal:~  x

Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/root/os-intro/template/presentation»...
remote: Enumerating objects: 111, done.
remote: Counting objects: 100% (111/111), done.
remote: Compressing objects: 100% (77/77), done.
remote: Total 111 (delta 42), reused 100 (delta 31), pack-reused 0 (from 0)
Получение объектов: 100% (111/111), 102.17 КиБ | 1.15 МБ/с, готово.
Определение изменений: 100% (42/42), готово.
Клонирование в «/root/os-intro/template/report»...
remote: Enumerating objects: 142, done.
remote: Counting objects: 100% (142/142), done.
remote: Compressing objects: 100% (97/97), done.
remote: Total 142 (delta 60), reused 121 (delta 39), pack-reused 0 (from 0)
Получение объектов: 100% (142/142), 341.09 КиБ | 1.91 МБ/с, готово.
Определение изменений: 100% (60/60), готово.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fca1d4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'
root@gayazovruzal:~#
```

## Перемещение между директориями

Удаляю лишние файлы с помощью утилиты `rm`, далее создаю необходимые каталоги используя `makefile` (рис. @fig:022).

```
root@gayazovruzal:~/work/study/2022-2023/Операционные системы
# ls
os-intro
root@gayazovruzal:~/work/study/2022-2023/Операционные системы
# cd os-intro
root@gayazovruzal:~/work/study/2022-2023/Операционные системы
/os-intro#
```

### *Удаление файлов и создание каталогов*

Добавляю все новые файлы для отправки на сервер с помощью команды git add и комментирую их с помощью git commit. Отправляю файлы на сервер с помощью git push.

## Выводы

При выполнении данной лабораторной работы я изучила идеологию и применение средств контроля версий, освоила умение по работе с git.

## Ответы на контрольные вопросы.

1. Системы контроля версий (VCS) - программное обеспечение для облегчения работы с изменяющейся информацией. Они позволяют хранить несколько версий изменяющейся информации, одного и того же документа, может предоставить доступ к более ранним версиям документа. Используется для работы нескольких человек над проектом, позволяет посмотреть, кто и когда внес какое-либо изменение и т. д. VCS применяются для: Хранения полной истории изменений, сохранения причин всех изменений, поиска причин изменений и совершивших изменение, совместной работы над проектами.
2. Хранилище – репозиторий, хранилище версий, в нем хранятся все документы, включая историю их изменения и прочей служебной информацией. commit – отслеживание изменений, сохраняет разницу в изменениях. История – хранит все изменения в проекте и позволяет при необходимости вернуться/обратиться к нужным данным. Рабочая копия – копия проекта, основанная на версии из хранилища, чаще всего последней версии.
3. Централизованные VCS (например: CVS, TFS, AccuRev) – одно основное хранилище всего проекта. Каждый пользователь копирует себе необходимые ему файлы из этого репозитория, изменяет, затем добавляет изменения обратно в хранилище. Децентрализованные VCS (например: Git, Bazaar) – у каждого пользователя свой вариант репозитория (возможно несколько вариантов), есть возможность добавлять и забирать изменения из любого репозитория. В отличие от классических, в распределенных (децентрализованных) системах контроля версий центральный репозиторий не является обязательным.

4. Сначала создается и подключается удаленный репозиторий, затем по мере изменения проекта эти изменения отправляются на сервер.
5. Участник проекта перед началом работы получает нужную ему версию проекта в хранилище, с помощью определенных команд, после внесения изменений пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются. К ним можно вернуться в любой момент.
6. Хранение информации о всех изменениях в вашем коде, обеспечение удобства командной работы над кодом.
7. Создание основного дерева репозитория: `git init`

Получение обновлений (изменений) текущего дерева из центрального репозитория: `git pull`

Отправка всех произведённых изменений локального дерева в центральный репозиторий: `git push`

Просмотр списка изменённых файлов в текущей директории: `git status`

Просмотр текущих изменений: `git diff`

Сохранение текущих изменений: добавить все изменённые и/или созданные файлы и/или каталоги: `git add` .

добавить конкретные изменённые и/или созданные файлы и/или каталоги: `git add имена_файлов`

удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории): `git rm имена_файлов`

Сохранение добавленных изменений:

сохранить все добавленные изменения и все изменённые файлы: `git commit -am 'Описание коммита'`

сохранить добавленные изменения с внесением комментария через встроенный редактор: `git commit`

создание новой ветки, базирующейся на текущей: `git checkout -b имя_ветки`

переключение на некоторую ветку: `git checkout имя_ветки` (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой)

отправка изменений конкретной ветки в центральный репозиторий: `git push origin имя_ветки`

слияние ветки с текущим деревом: `git merge --no-ff имя_ветки`

Удаление ветки:

удаление локальной уже слитой с основным деревом ветки: `git branch -d имя_ветки`

принудительное удаление локальной ветки: `git branch -D имя_ветки`

удаление ветки с центрального репозитория: `git push origin :имя_ветки`

8. `git push -all` отправляем из локального репозитория все сохраненные изменения в центральный репозиторий, предварительно создав локальный репозиторий и сделав предварительную конфигурацию.
9. Ветвление - один из параллельных участков в одном хранилище, исходящих из одной версии, обычно есть главная ветка. Между ветками, т. е. их концами возможно их слияние. Используются для разработки новых функций.
10. Во время работы над проектом могут создаваться файлы, которые не следуют добавлять в репозиторий. Например, временные файлы. Можно прописать шаблоны игнорируемых при добавлении в репозиторий типов файлов в файл `.gitignore` с помощью сервисов.

## Список литературы

1. Лабораторная работа № 2 [Электронный ресурс] URL: <https://esystem.rudn.ru/mod/page/view.php?id=970819>