

# **Отчёт по лабораторной работе №2**

**Операционные системы**

Серёгина Ирина Андреевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
2.1	Установка ПО . . . . .	6
2.2	Базовая настройка git . . . . .	6
2.3	Создание ssh-ключа . . . . .	7
2.4	Создание pgp-ключа . . . . .	8
2.5	Регистрация на github . . . . .	9
2.6	Добавление gpg-ключа . . . . .	10
2.7	Настройка автоматических подписей коммитов git . . . . .	10
2.8	Настройка gh . . . . .	11
2.9	Создание репозитория курса на основе шаблона . . . . .	12
2.10	Настройка каталога курса . . . . .	13
<b>3</b>	<b>Ответы на контрольные вопросы</b>	<b>15</b>
<b>4</b>	<b>Выводы</b>	<b>17</b>
	<b>Список литературы</b>	<b>18</b>

# Список иллюстраций

2.1	установка git и gh . . . . .	6
2.2	установка имени и почты владельца . . . . .	6
2.3	Настройка utf-8 . . . . .	6
2.4	Задаю имя начальной ветки . . . . .	7
2.5	Задаю параметры autocrlf, safecrlf . . . . .	7
2.6	Алгоритм rsa . . . . .	7
2.7	Алгоритм ed25519 . . . . .	8
2.8	Генерирую pgr-ключ . . . . .	8
2.9	Защищаю pgr-ключ . . . . .	9
2.10	Аккаунт на github . . . . .	9
2.11	Копирую отпечаток приватного ключа . . . . .	10
2.12	Копирую ключ . . . . .	10
2.13	Добавляю ключ . . . . .	10
2.14	Настройка коммитов . . . . .	11
2.15	Авторизация . . . . .	11
2.16	Завершение авторизации . . . . .	12
2.17	Создание репозитория . . . . .	12
2.18	Клонирование репозитория . . . . .	13
2.19	Каталог курса . . . . .	13
2.20	Удаление лишних элементов . . . . .	13
2.21	Создание необходимых каталогов . . . . .	14
2.22	Добавление данных . . . . .	14
2.23	Отправка данных . . . . .	14

## Список таблиц

# 1 Цель работы

Цель работы заключается в изучении идеологии и применении средств контроля версий, освоении умения по работе с git.

#Задания 1.Установка ПО 2.Базовая настройка git 3.Создание ssh-ключа 4.Создание pgr-ключа 5.Регистрация на github 6.Добавление grpg-ключа 7.Настройка автоматических подписей коммитов git 8.Настройка gh 9.Создание репозитория курса на основе шаблона 10.Настройка каталога курса

## 2 Выполнение лабораторной работы

### 2.1 Установка ПО

Я захожу в роль супер-пользователя и устанавливаю git, gh (рис. 2.1).

```
[irina@fedora ~]$ sudo dnf -y install git
[sudo] пароль для irina:
Последняя проверка окончания срока действия метаданных: 2:17:14 назад, Сб 18 фев 2023 12:38:23.
Пакет git-2.39.1-1.fc36.x86_64 уже установлен.
Зависимости разрешены.
Отсутствуют действия для выполнения.
Выполнено!
[irina@fedora ~]$ sudo dnf -y install gh
Последняя проверка окончания срока действия метаданных: 2:17:41 назад, Сб 18 фев 2023 12:38:23.
```

Рис. 2.1: установка git и gh

### 2.2 Базовая настройка git

Задаю имя и электронную почту владельца репозитория (рис. 2.2).

```
[irina@fedora ~]$ git config --global user.name "Irina Seregina"
[irina@fedora ~]$ git config --global user.email "isereguine@yandex.ru"
```

Рис. 2.2: установка имени и почты владельца

Настраиваю utf-8 в выводе сообщений git (рис. 2.3).

```
[irina@fedora ~]$ git config --global core.quotepath false
```

Рис. 2.3: Настройка utf-8

Задаю имя начальной ветки master (рис. 2.4).

```
[irina@fedora ~]$ git config --global init.defaultBranch master
```

Рис. 2.4: Задаю имя начальной ветки

Задаю параметры autocrlf, safecrlf (рис. 2.5).

```
[irina@fedora ~]$ git config --global core.autocrlf input
[irina@fedora ~]$ git config --global core.safecrlf warn
```

Рис. 2.5: Задаю параметры autocrlf, safecrlf

## 2.3 Создание ssh-ключа

Создаю ключ по алгоритму rsa с ключем размером 4096 бит (рис. 2.6).

```
[irina@fedora ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/irina/.ssh/id_rsa):
Created directory '/home/irina/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/irina/.ssh/id_rsa
Your public key has been saved in /home/irina/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:VG3Qxlrwlp+KWpI2/+R49MGj46QLfN/bUXp/2bKPLO irina@fedora
The key's randomart image is:
+---[RSA 4096]---+
|                 o= . |
|                . B . |
|               . = o . |
|              . . =.. |
|             S   +.o. |
|            +. +oo+ |
|           * + oo= |
|          *.++.& |
|         .E*B@* |
+-----[SHA256]-----+
[irina@fedora ~]$
```

Рис. 2.6: Алгоритм rsa

Создаю ключ по алгоритму ed25519 (рис. 2.7).

```

[irina@fedora ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/irina/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/irina/.ssh/id_ed25519
Your public key has been saved in /home/irina/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:gyIIwckJ8QACXlneikE77saJGVH+b/B0U+N7P5R6z8c irina@fedora
The key's randomart image is:
+--[ED25519 256]--+
|/..+..          |
|.X..           |
|.o. .          |
|oo.. . o       |
|o.+..o . So . . |
|. + ooo. o.. o  |
|+o= . = . . . o |
|o* . + . o o.E  |
|o. . . . o.o+   |
+-----[SHA256]-----+
[irina@fedora ~]$

```

Рис. 2.7: Алгоритм ed25519

## 2.4 Создание ргр-ключа

Генерирую ргр-ключ, из опций выбираю те, которые были представлены в инструкции по лабораторной работе (рис. 2.8).

```

[irina@fedora ~]$ gpg --full-generate-key
gpg (GnuPG) 2.3.7; 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: создан каталог '/home/irina/.gnupg'
gpg: создан щит с ключами '/home/irina/.gnupg/pubring.kbx'
Выберите тип ключа:
  (1) RSA and RSA
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (9) ECC (sign and encrypt) *default*
 (10) ECC (только для подписи)
 (14) Existing key from card
Ваш выбор? 1

```

Рис. 2.8: Генерирую ргр-ключ

Защищаю ключ с помощью пароля (рис. 2.9).



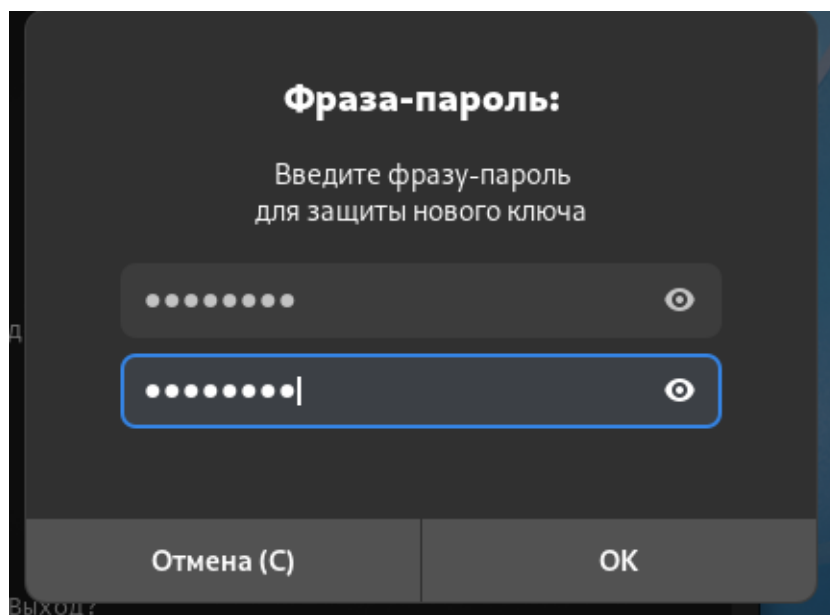


Рис. 2.9: Защищаю pgr-ключ

## 2.5 Регистрация на github

В прошлом семестре я уже создала аккаунт на github, поэтому регистрация не требуется (рис. 2.10).

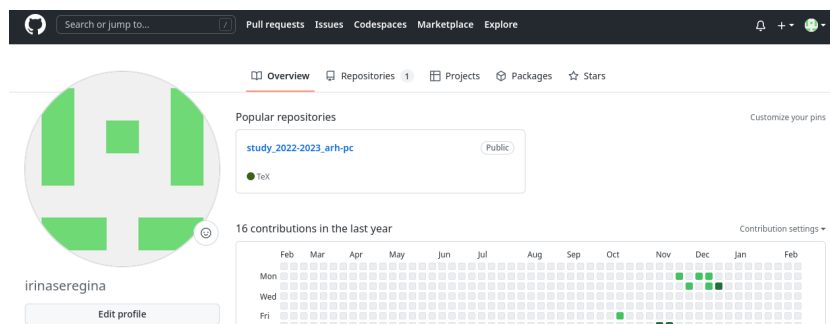


Рис. 2.10: Аккаунт на github

## 2.6 Добавление gpg-ключа

Вывожу список ключей и копирую отпечаток приватного ключа (рис. 2.11).

```
irina@fedora ~]$ gpg --list-secret-keys --keyid-format LONG
gpg: проверка таблицы доверия
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: глубина: 0 достоверных: 1 подписанных: 0 доверие: 0-, 0q, 0n, 0m, 0f, 1u
/home/irina/.gnupg/pubring.kbx
-----
sec rsa4096/EFB5DEDE05572AFD 2023-02-18 [SC]
    431A3950FC17A67994C67DC9EFB5DEDE05572AFD
uid [ абсолютно ] SereginaIrina <isereguine@yandex.ru>
ssb rsa4096/09EC9046ED37C792 2023-02-18 [E]
```

Рис. 2.11: Копирую отпечаток приватного ключа

Копирую мой ключ в буфер обмена с помощью xclip (рис. 2.12).

```
irina@fedora ~]$ gpg --armor --export EFB5DEDE05572AFD | xclip -set clip
```

Рис. 2.12: Копирую ключ

Перехожу на сайт и добавляю новый ключ, куда вставляю его из буфера обмена (рис. 2.13).

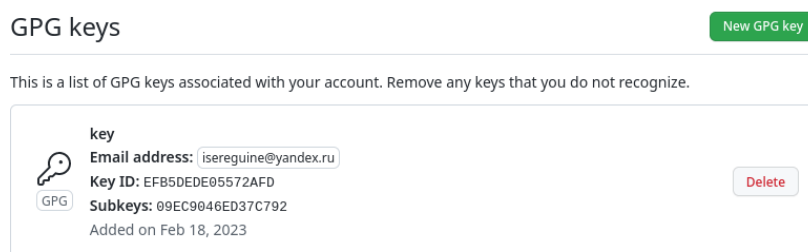


Рис. 2.13: Добавляю ключ

## 2.7 Настройка автоматических подписей коммитов git

Настраиваю коммиты так, чтобы при их подписи использовалась моя почта (рис. 2.14).

```
[irina@fedora ~]$ git config --global user.signinkey EFB5DEDE05572AFD
[irina@fedora ~]$ git config --global commit.gpgsign true
[irina@fedora ~]$ git config --global gpg.program $(which gpg2)
[irina@fedora ~]$
```

Рис. 2.14: Настройка коммитов

## 2.8 Настройка gh

Я авторизируюсь, отвечая на несколько вопросов, после этого выбираю продолжить авторизацию в браузере (рис. 2.15).

```
[irina@fedora ~]$ 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? Login with a web browser

! First copy your one-time code: 06D5-EB63
Press Enter to open github.com in your browser...
✓ Authentication complete.
- gh config set -h github.com git_protocol https
✓ Configured git protocol
✓ Logged in as irinaseregina
```

Рис. 2.15: Авторизация

Авторизация прошла успешно (рис. 2.16).

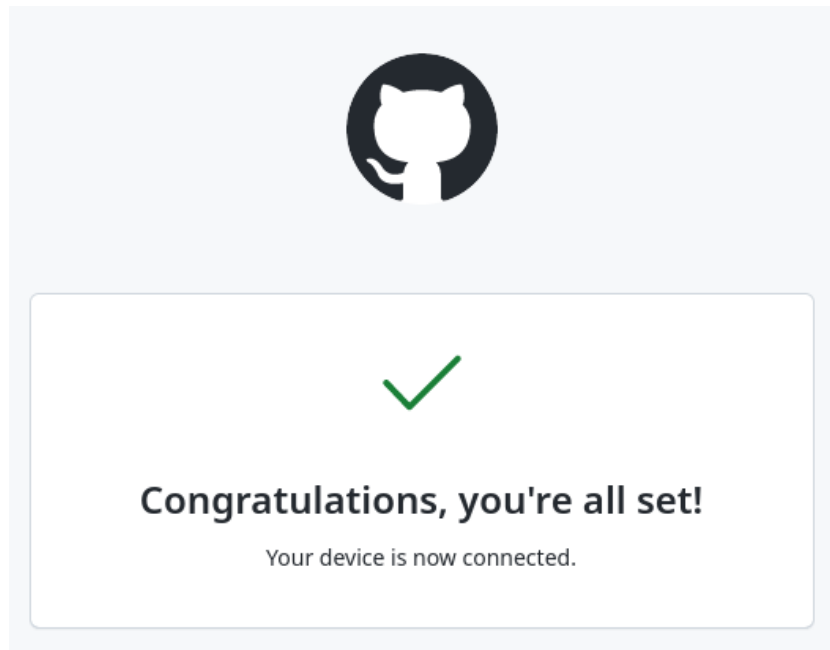


Рис. 2.16: Завершение авторизации

## 2.9 Создание репозитория курса на основе шаблона

Создаю каталог курса с помощью `mkdir`, после этого перехожу в каталог курса и создаю репозиторий на основе шаблона (рис. 2.17).

```
irin@fedora ~]$ mkdir -p ~/work/study/2022-2023/"Операционные системы"
irin@fedora ~]$ cd ~/work/study/2022-2023/"Операционные системы"
irin@fedora Операционные системы]$ gh repo create study_2022-2023_os-intro --template=yamadharma/course-d
irectory-student-template --public
✓ Created repository irinaseregina/study_2022-2023_os-intro on GitHub
```

Рис. 2.17: Создание репозитория

Затем клонирую репозиторий себе в директорию (рис. 2.18).

```
[irina@fedora Операционные системы]$ git clone --recursive https://github.com/irinaseregina/study_2022-2023
os-intro.git os-intro
Клонирование в «os-intro»...
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
Получение объектов: 100% (27/27), 16.93 КиБ | 101.00 КиБ/с, готово.
Определение изменений: 100% (1/1), готово.
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) зареги-
стрирован по пути «template/report»
Клонирование в «/home/irina/work/study/2022-2023/Операционные системы/os-intro/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
Получение объектов: 100% (82/82), 92.90 КиБ | 91.00 КиБ/с, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/home/irina/work/study/2022-2023/Операционные системы/os-intro/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
Получение объектов: 100% (101/101), 327.25 КиБ | 242.00 КиБ/с, готово.
Определение изменений: 100% (40/40), готово.
Submodule path 'template/presentation': checked out 'b1be3800ee91f5809264cb755d316174540b753e'
Submodule path 'template/report': checked out '1d1b61dcac9c287a83917b82e3aef11a33b1e3b2'
[irina@fedora Операционные системы]$
```

Рис. 2.18: Клонирование репозитория

## 2.10 Настройка каталога курса

Перехожу в каталог курса, смотрю содержимое с помощью ls (рис. 2.19).

```
[irina@fedora Операционные системы]$ cd os-intro
[irina@fedora os-intro]$ ls
CHANGELOG.md  COURSE  Makefile  README.en.md  README.md
config        LICENSE package.json  README.git-flow.md  template
[irina@fedora os-intro]$
```

Рис. 2.19: Каталог курса

Удаляю лишние файлы с помощью rm package.json, проверяю, удалились ли они (рис. 2.20).

```
[irina@fedora os-intro]$ rm package.json
[irina@fedora os-intro]$ ls
CHANGELOG.md  config  COURSE  LICENSE  Makefile  README.en.md  README.git-flow.md  README.md  template
[irina@fedora os-intro]$
```

Рис. 2.20: Удаление лишних элементов

Создаю необходимые каталоги (рис. 2.21).

```

[irina@fedora os-intro]$ echo os-intro > COURSE
[irina@fedora os-intro]$ make
[irina@fedora os-intro]$ ls
CHANGELOG.md  COURSE  LICENSE  prepare  project-personal  README.git-flow.md  template
config        labs   Makefile  presentation  README.en.md      README.md

```

Рис. 2.21: Создание необходимых каталогов

С помощью `git add .`, `git commit` добавляю файлы, которые нужно отправить, и добавляю комментарии к ним (рис. 2.22).

```

[irina@fedora os-intro]$ git add .
[irina@fedora os-intro]$ git commit -am 'feat(main): make course structure'
error: gpg не удалось подписать данные
fatal: сбой записи объекта коммита
[irina@fedora os-intro]$ git commit -am 'feat(main): make course structure'
error: gpg не удалось подписать данные
fatal: сбой записи объекта коммита
[irina@fedora os-intro]$ git config --global --unset commit.gpgsign
[irina@fedora os-intro]$ git commit -am 'feat(main): make course structure'
[master c0312ed] feat(main): make course structure
361 files changed, 100327 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
create mode 100644 labs/lab01/presentation/presentation.md

```

Рис. 2.22: Добавление данных

С помощью `git push` отправляю данные на сервер (рис. 2.23).

```

[irina@fedora os-intro]$ git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 342.40 КиБ | 1.05 МБ/с, готово.
Всего 38 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To https://github.com/irinaseregina/study_2022-2023_os-intro.git
 f61bfff..c0312ed master -> master
[irina@fedora os-intro]$

```

Рис. 2.23: Отправка данных

### 3 Ответы на контрольные вопросы

1. VCS - системы контроля версий, ПО, облегчающее работу с изменяемой информацией, которое позволяет одновременно хранить несколько версий одного элемента, работать над информацией совместно и отслеживать чужие поправки.
2. Хранилище - часть, в которой хранятся версии информации и данные о её изменении, коммит - фиксатор изменений, позволяющий их отслеживать, история - то, что позволяет вернуться к более ранней версии данных, хранит историю изменений, рабочая копия - в основном последняя копия документа, основанная на копии из хранилища.
3. Централизованные VCS отличаются от децентрализованных тем, что у них единый общий репозиторий, из которого каждый пользователь может брать информацию, а у децентрализованных каждый имеет свою версию репозитория, имея возможность брать данные из репозитория других пользователей. К первым относится TFS, ко вторым - Git.
4. Создание и подключение репозитория, а затем постепенная отправка данных на сервер.
5. Для начала мы берем изначальную версию данных, а после изменения загружаем на сервер новую версию, при этом старая версия не удаляется с сервера.
6. Обеспечение удобной совместной работы и постоянного доступа к данным о состоянии и изменениях в данных.
7. `git init` - создание основного дерева репозитория `git pull` - обновление теку-

щего дерева из центрального репозитория `git push` - отправка изменений локального дерева на сервер `git status` - просмотр списка измененных файлов в текущей директории `git diff` - просмотр текущих изменений `git add .` - добавление созданных и измененных файлов и каталогов `git commit -m ""` - сохранение добавленных изменений с комментарием `git checkout -b ""` - создание ветки с именем `git branch -d ""` - удаление ветки с именем

8. При использовании команды `git pull` мы загружаем информацию из удаленного репозитория и обновляем локальный.
9. Ветвление означает создание параллельных веток для более удобной совместной работы над проектом, грубо говоря это история изменений. Изначально существует одна главная ветка.
10. Не все файлы требуют добавления в репозиторий, они могут просто быть бесполезными, поэтому с помощью `.gitignore` их можно проигнорировать при добавлении информации на сервер.



## 4 Выводы

Я изучила идеологию и применение средств контроля версий, освоила умения по работе с git.

## **Список литературы**