

# **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

**Система контроля версий git**

Шурыгин Илья Максимович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Контрольные вопросы:</b>	<b>13</b>
<b>5</b>	<b>Выводы</b>	<b>15</b>

## Список иллюстраций

3.1	Генерируем ключ . . . . .	7
3.2	Настраиваем ключ . . . . .	8
3.3	Настраиваем ключ . . . . .	8
3.4	Копируем отпечаток приватного ключа . . . . .	8
3.5	Скопируйте ключ в буфер обмена . . . . .	9
3.6	Ключ на GitHub . . . . .	9
3.7	Настройка автоматических подписей . . . . .	9
3.8	Авторизация . . . . .	10
3.9	Авторизация . . . . .	10
3.10	Создание репозитория курса . . . . .	11
3.11	Копируем шаблон с GitHub . . . . .	11
3.12	Создание репозитория курса . . . . .	11
3.13	Копируем шаблон с GitHub . . . . .	12

## **Список таблиц**

# 1 Цель работы

Целью моей работы является: изучить идеологию и применение средств контроля версий и освоить умения по работе с git.

## 2 Задание

Необходимо создать базовую конфигурацию для работы с git, создать ключ SSH и PGP, настроить подписи git и создать локальный каталог для выполнения заданий по предмету.

### 3 Выполнение лабораторной работы

1. Аккаунт на Github у меня уже создан и настроен, ключ SSH настроен. Создадим ключ pgp.(рис. 3.1)(рис. 3.2)(рис. 3.3)

```
[imshurihgin@fedora ~]$ gpg --full-generate-key
gpg (GnuPG) 2.3.4; 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.

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
длина ключей RSA может быть от 1024 до 4096.
Какой размер ключа Вам необходим? (3072) 4096
Запрошенный размер ключа - 4096 бит
Выберите срок действия ключа.
  0 = не ограничен
  (30) = срок действия ключа - 30 дней
```

Рис. 3.1: Генерируем ключ

```

длина ключей RSA может быть от 1024 до 4096.
Какой размер ключа Вам необходим? (3072) 4096
Запрошенный размер ключа - 4096 бит
Выберите срок действия ключа.
    0 = не ограничен
    <n> = срок действия ключа - n дней
    <n>w = срок действия ключа - n недель
    <n>m = срок действия ключа - n месяцев
    <n>y = срок действия ключа - n лет
Срок действия ключа? (0) 0
Срок действия ключа не ограничен
Все верно? (y/N) y

GnuPG должен составить идентификатор пользователя для идентификации ключа

Ваше полное имя: imshurihgin
Адрес электронной почты: bicyclestandard534@gmail.com
Примечание:
Вы выбрали следующий идентификатор пользователя:
    "imshurihgin <bicyclestandard534@gmail.com>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? o
Необходимо получить много случайных чисел. Желательно, чтобы Вы

```

Рис. 3.2: Настраиваем ключ

```

случайных чисел больше возможностей получить достаточное количество энтропии
gpg: /home/imshurihgin/.gnupg/trustdb.gpg: создана таблица доверия
gpg: создан каталог '/home/imshurihgin/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/home/imshurihgin/.gnupg/openpgp-revocs.d/9F8198E08316FDC8AB1C816F15483DAB7ED.rev'.
открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2023-02-16 [SC]
      00C19F8198E08316FDC8AB1C816F15483DAB7ED
uid    imshurihgin <bicyclestandard534@gmail.com>
sub   rsa4096 2023-02-16 [E]

```

Рис. 3.3: Настраиваем ключ

2. Выводим список ключей и копируем отпечаток приватного ключа, далее скопируем ваш сгенерированный PGP ключ в буфер обмена. Вставим полученный ключ в GitHub.(рис. 3.4)(рис. 3.5)(рис. 3.6)

```

[imshurihgin@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/imshurihgin/.gnupg/pubring.kbx
-----
sec   rsa4096/C816F15483DAB7ED 2023-02-16 [SC]
      00C19F8198E08316FDC8AB1C816F15483DAB7ED
uid    [ абсолютно ] imshurihgin <bicyclestandard534@gmail.com>
ssb   rsa4096/E9568034AC68D2B2 2023-02-16 [E]

```

Рис. 3.4: Копируем отпечаток приватного ключа



```
[imshurihgin@fedora ~]$ gpg --armor --export C816F15483DAB7ED | xclip -sel clip
bash: xclip: команда не найдена...
Установить пакет «xclip», предоставляющий команду «xclip»? [N/y] y

* Ожидание в очереди...
Следующие пакеты должны быть установлены:
xclip-0.13-16.git11cba61.fc36.x86_64  Command line clipboard grabber
Продолжить с этими изменениями? [N/y] y

* Ожидание в очереди...
* Ожидание аутентификации...
* Ожидание в очереди...
* Загрузка пакетов...
* Запрос данных...
* Проверка изменений...
* Установка пакетов...

[imshurihgin@fedora ~]$ gpg --armor --export C816F15483DAB7ED | xclip -sel clip
[imshurihgin@fedora ~]$
```

Рис. 3.5: Скопируйте ключ в буфер обмена

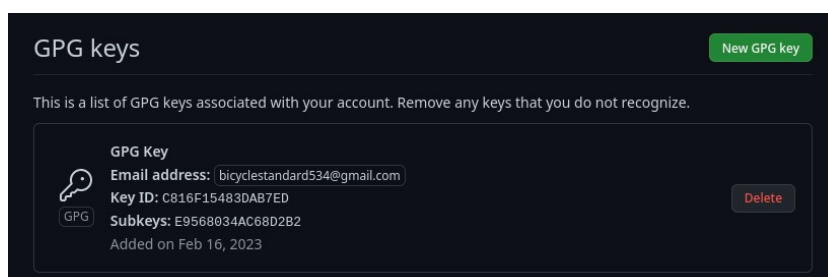


Рис. 3.6: Ключ на GitHub

## 2. Настроим автоматические подписи коммитов git.(рис. 3.7)

```
[imshurihgin@fedora ~]$ gpg --armor --export C816F15483DAB7ED | xclip -sel clip
[imshurihgin@fedora ~]$ git config --global user.signingkey C816F15483DAB7ED
[imshurihgin@fedora ~]$ git config --global commit.gpgsign true
[imshurihgin@fedora ~]$ git config --global gpg.program $(which gpg2)
[imshurihgin@fedora ~]$
```

Рис. 3.7: Настройка автоматических подписей

## 3. Настроим gh. Для этого необходимо авторизоваться.(рис. 3.8)(рис. 3.9)

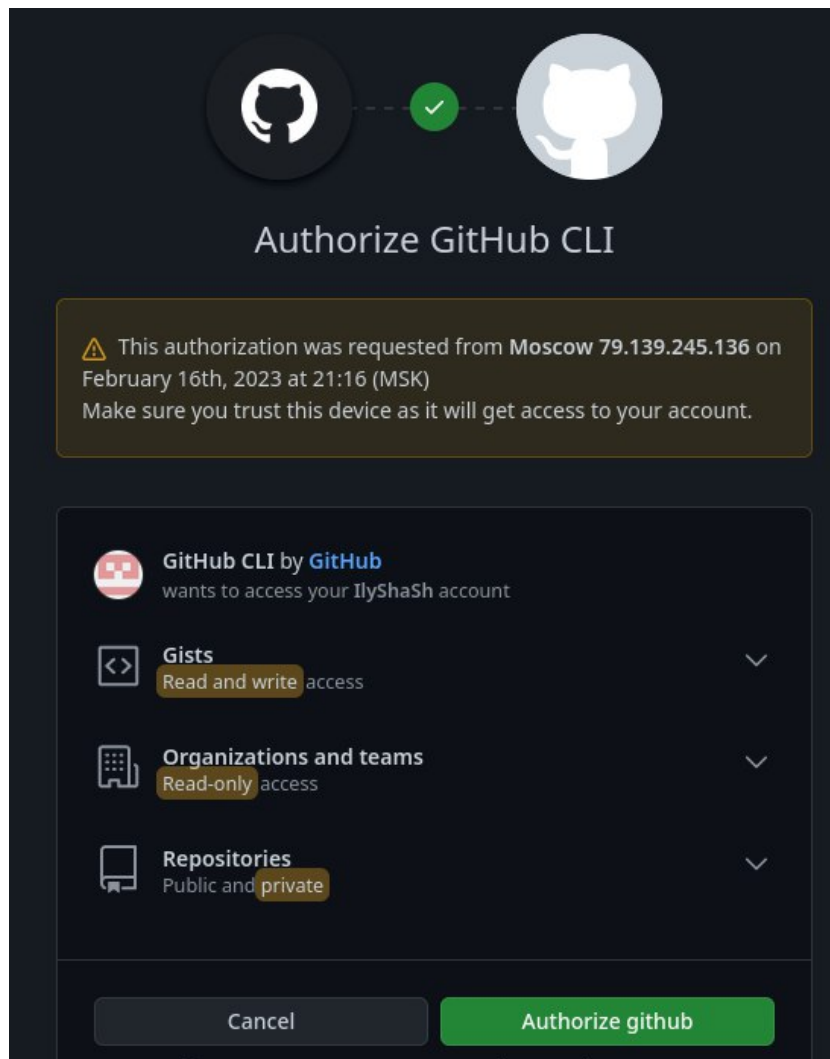


Рис. 3.8: Авторизация

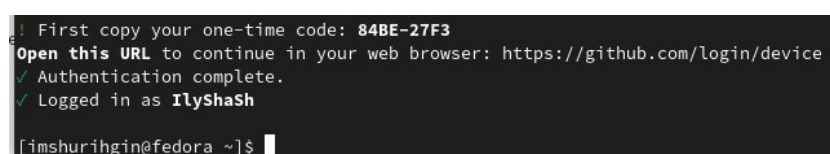


Рис. 3.9: Авторизация

4. Создадим репозиторий курса на основе шаблона. Для этого скопируем шаблон с GitHub.(рис. 3.10)(рис. 3.11)

```
[imshurighin@fedora ~]$ mkdir -p ~/work/study/2022-2023/"Операционные системы"
[imshurighin@fedora ~]$ cd ~/work/study/2022-2023/"Операционные системы"
[imshurighin@fedora Операционные системы]$ gh repo create study_2022-2023_os-intro --template=yamadharma/course-directory-student-template --public
/ Created repository IlyShash/study_2022-2023_os-intro on GitHub
[imshurighin@fedora Операционные системы]$ git clone --recursive git@github.com:owner>/study_2022-2023_os-intro.git os-intro
bash: owner: Нет такого файла или каталога
[imshurighin@fedora Операционные системы]$ git clone --recursive git@github.com:IlyShash/study_2022-2023_os-intro.git os-intro
```

Рис. 3.10: Создание репозитория курса

```
[imshurighin@fedora Операционные системы]$ git clone --recursive git@github.com:IlyShash/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 | 4.23 Миб/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) зарегистрирован по пути «temp
late/report»
Клонирование в «/home/imshurighin/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 | 318.00 Миб/с, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/home/imshurighin/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.
```

Рис. 3.11: Копируем шаблон с GitHub

5. Настроим каталог курса: удалим лишние файлы, создадим необходимые каталоги и отправим файлы на сервер.(рис. 3.12)(рис. 3.13)

```
[imshurighin@fedora Операционные системы]$ cd ~/work/study/2022-2023/"Операционные системы"/os-intro
[imshurighin@fedora os-intro]$ rm package.json
[imshurighin@fedora os-intro]$ echo os-intro > COURSE
[imshurighin@fedora os-intro]$ make
[imshurighin@fedora os-intro]$
```

Рис. 3.12: Создание репозитория курса

```

create mode 100644 project-personal/stage6/report/bib/cite.bib
create mode 100644 project-personal/stage6/report/image/placeimg_800_600_
create mode 100644 project-personal/stage6/report/pandoc/csl/gost-r-7-0-5
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_e
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_f
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_s
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_t
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxn
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxn
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxn
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxn
create mode 100644 project-personal/stage6/report/report.md
[imshurihgin@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), 343.05 КиБ | 1.55 МиБ/с, готово.
Всего 38 (изменений 4), повторно использовано 0 (изменений 0), повторно ис
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:IlyShash/study_2022-2023_os-intro.git
   3665d89..ec4fela  master -> master
[imshurihgin@fedora os-intro]$

```

Рис. 3.13: Копируем шаблон с GitHub

## 4 Контрольные вопросы:

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?
  - система, позволяющая работать нескольким людям над одним проектом.
2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.
  - хранилище (репозиторий) - директория, хранящая конкретный проект; коммит - текущее состояние рабочей копии; история - последовательность коммитов в порядке, в котором они добавлялись в репозиторий; рабочая копия - текущее состояние репозитория, которое находится в состоянии изменения.
3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.
  - в централизованных VCS (Mercurial) все пользователи подключены к единому серверу; в децентрализованных VCS пользователи подключены к нескольким владельцам.
4. Опишите действия с VCS при единоличной работе с хранилищем.
  - при единоличной работе с хранилищем все изменения, созданные пользователем, не влияют на общий репозиторий.

5. Опишите порядок работы с общим хранилищем VCS.
  - из общего хранилища можно получать изменения проекта.
6. Каковы основные задачи, решаемые инструментальным средством git?
  - git позволяет нескольким людям работать над одним проектом.
7. Назовите и дайте краткую характеристику командам git.
  - add - добавить файлы в коммит; push - отправить коммит на удалённый репозиторий; pull - импортировать проект с удалённого репозитория.
8. Приведите примеры использования при работе с локальным и удалённым репозиториями.
  - работа над интернет-магазином команды разработчиков
9. Что такое и зачем могут быть нужны ветви (branches)?
  - создав новую ветвь, можно, не вредя проекту, работать над конкретной частью проекта.
10. Как и зачем можно игнорировать некоторые файлы при commit?
  - некоторые файлы вполне могут быть специфичными для пользователя.

## 5 Выводы

Вывод: я создал новый репозитория курса, в котором настроил ключи SSH и PGP. Теперь смогу контролировать версии своих проектов.