# Отчет по лабораторной работе №2

**Первоначальна настройка git**

Акопян Сатеник Манвеловна

# Содержание

# Список иллюстраций

# Список таблиц

# 1 Цель работы

Изучить идеологию и применение средств контроля версий.

Освоить умения по работе с git.

# 2 Выполнение лабораторной работы

1.Установка программного обеспечения

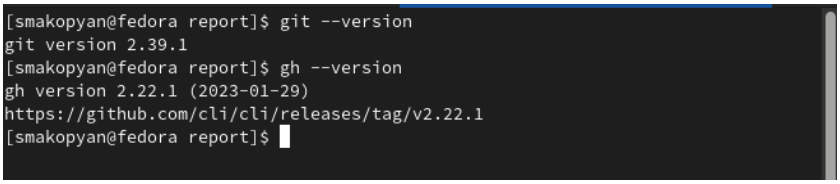


Рис. 2.1: рисунок 1

2.Базовая настройка git.Сделаем предварительную конфигурацию git, а так же настроим utf-8 в выводе сообщений git и зададим имя начальной ветки (будем называть её master)

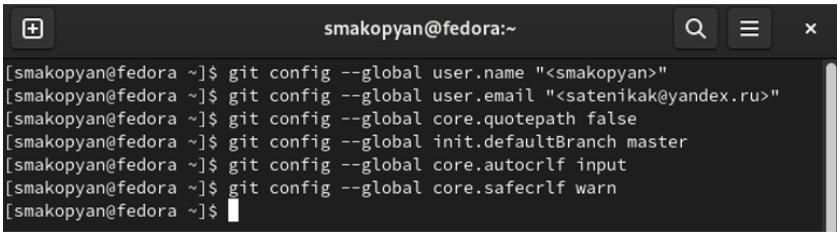


Рис. 2.2: рисунок 2

3.Создание ключей ssh.


Рис. 2.3: рисунок 3

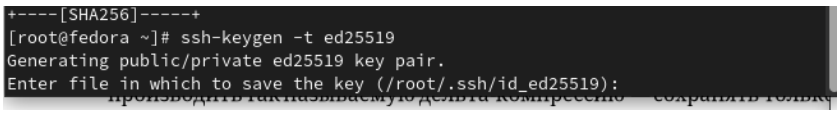


Рис. 2.4: рисунок 4

## 4.Создание ключей pgp

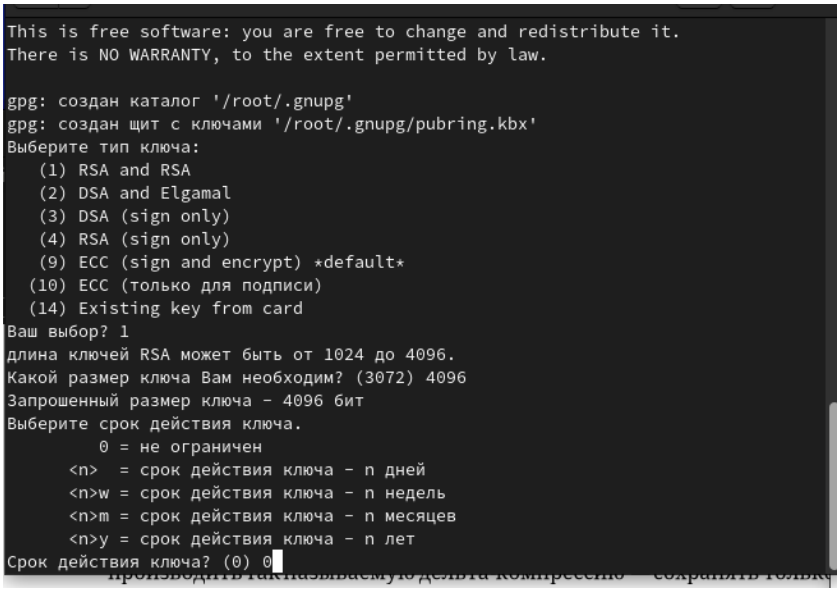


Рис. 2.5: рисунок 5

## 5.Настройка github

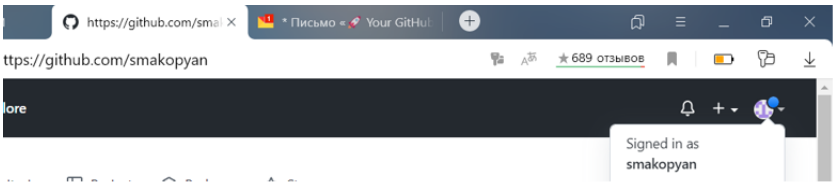


Рис. 2.6: рисунок 6

## 6.Добавление PGP ключа в GitHub

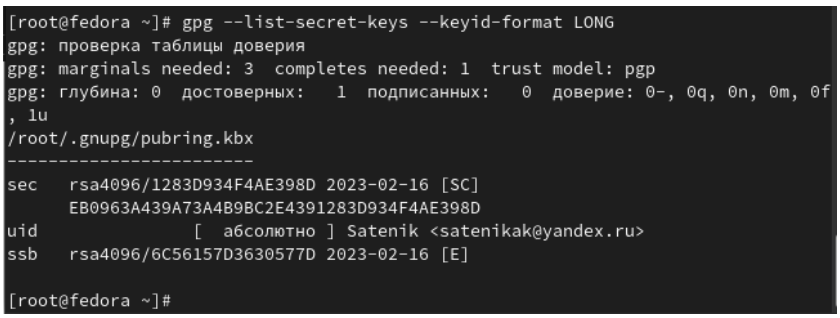


Рис. 2.7: рисунок 7

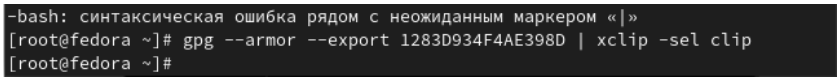


Рис. 2.8: рисунок 8

7.Настройка автоматических подписей коммитов git

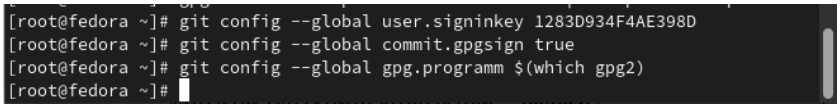


Рис. 2.9: рисунок 9

8.Настройка gh

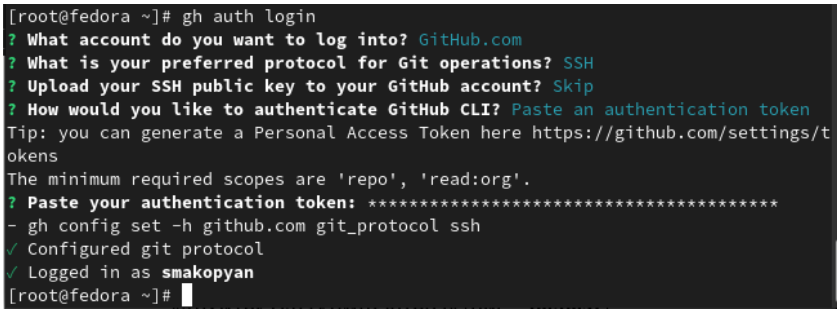


Рис. 2.10: рисунок 10

9.Создание репозитория курса на основе шаблона

```
[smakopyan@fedora Операционные системы]$ mkdir -p ~/work/study/2022-2023/"Операц
ионные системы"
[smakopyan@fedora Операционные системы]$ cd ~/work/study/2022-2023/"Операционные
 системы"
[smakopyan@fedora Операционные системы]$ gh repo create study_2022-2023_os-intro
 --template=yamadharma/course-directory-student-template --public
✓ Created repository smakopyan/study_2022-2023_os-intro on GitHub
[smakopyan@fedora Операционные системы]$ git clone --recursive git@github.com:<o
wner>/study_2022-2023_os-intro.git os-intro
bash: owner: Нет такого файла или каталога
[smakopyan@fedora Операционные системы]$ git clone --recursive git@github.com:sm
akopyan/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 | 8.47 МиБ/с, готово.
Определение изменений: 100% (1/1), готово.
```

Рис. 2.11: рисунок 11

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

Рис. 2.12: рисунок 12

```
[smakopyan@fedora os-intro]$ git add .
[smakopyan@fedora os-intro]$ git commit -am 'a'
[master 4068b12] a
 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
 create mode 100644 labs/lab01/report/Makefile
 create mode 100644 labs/lab01/report/bib/cite.bib
```

# Контрольные вопросы

1.Что такое системы контроля версий (VCS) и для решения каких задач они предназна
Системы контроля версий (Version Control System, VCS) применяются при работе неск

2.Объясните следующие понятия VCS и их отношения: хранилище, commit, история, раб
Репозиторий - хранилище версий - в нем хранятся все документы вместе с историей и

изменения и другой служебной информацией

commit -редактирует репозиторий, заносит туда изменения

Рабочая копия - копия проекта, связанная с репозиторием

3.Что представляют собой и чем отличаются централизованные и децентрализованные V

Централизованные VCS :одно основное хранилище всего проекта, каждый пользователь
Децентрализованные VCS: У каждого пользователя свой вариант (возможно не

один) репозитория, присутствует возможность добавлять и забирать изменения из любого репозитория

4.Опишите действия с VCS при единоличной работе с хранилищем.
Сначала создаем и подключаем удаленный репозиторий, затем по мере изменения проек

5.Опишите порядок работы с общим хранилищем VCS.
Пользователь перед началом работы получает нужную ему версию файлов, после внесен

6.Каковы основные задачи, решаемые инструментальным средством git?
Основными задачами, решаемыми инструментальным средством git являются: хранение и

7.Назовите и дайте краткую характеристику командам git.
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 ос

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 буде

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.Приведите примеры использования при работе с локальным и удалённым репозиторием

9.Что такое и зачем могут быть нужны ветви (branches)?

Ветви нужны для совместной работы над проектом.

10.Как и зачем можно игнорировать некоторые файлы при commit?

Т.к. во время работы над проектом могут создаваться файлы, которые не должны попа

# 3 Выводы

В результате данной лабораторной работы, мы изучили идеологию и применение средств контроля версий, а также освоили умения по работе с git.

# Список литературы