

# **Отчёт по лабораторной работе №2**

**Управление версиями**

Фарида Гулиева НКНбд 01-21

# Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	11
4	Контрольные вопросы	12
	Список литературы	16

# List of Figures

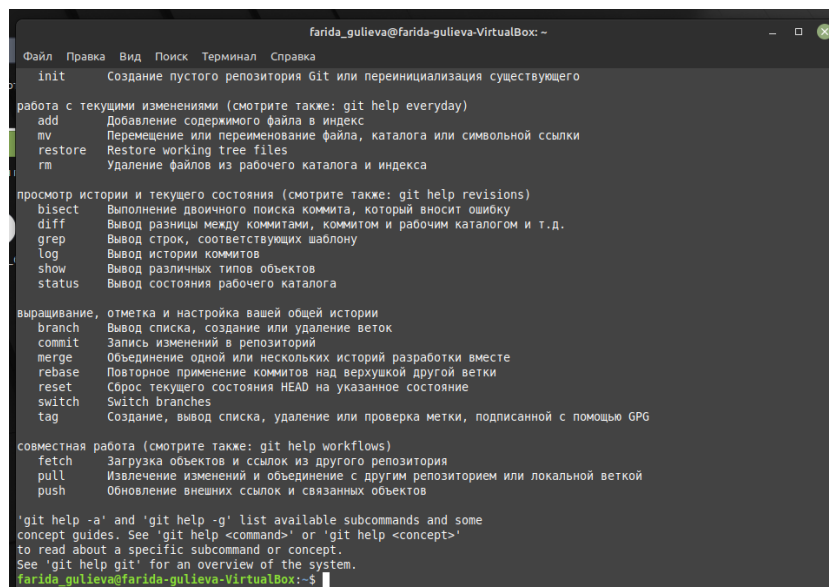
2.1	Загрузка пакетов . . . . .	5
2.2	Параметры репозитория . . . . .	6
2.3	rsa-4096 . . . . .	6
2.4	ed25519 . . . . .	7
2.5	GPG ключ . . . . .	7
2.6	GPG ключ . . . . .	8
2.7	Параметры репозитория . . . . .	8
2.8	Связь репозитория с аккаунтом . . . . .	9
2.9	Загрузка шаблона . . . . .	9
2.10	Первый коммит . . . . .	10

# 1 Цель работы

Целью данной работы является изучение идеологии и применения средств контроля версий и освоение умений работать с git.

## 2 Выполнение лабораторной работы

Устанавливаем git, git-flow и gh.



```
farida_gulieva@farida-gulieva-VirtualBox ~
Файл Правка Вид Поиск Терминал Справка
init      Создание пустого репозитория Git или переинициализация существующего

работа с текущими изменениями (смотрите также: git help everyday)
add       Добавление содержимого файла в индекс
mv        Перемещение или переименование файла, каталога или символической ссылки
restore   Restore working tree files
rm        Удаление файлов из рабочего каталога и индекса

просмотр истории и текущего состояния (смотрите также: git help revisions)
bisect    Выполнение двоичного поиска коммита, который вносит ошибку
diff      Вывод разницы между коммитами, коммитом и рабочим каталогом и т.д.
grep      Вывод строк, соответствующих шаблону
log       Вывод истории коммитов
show      Вывод различных типов объектов
status    Вывод состояния рабочего каталога

выращивание, отметка и настройка вашей общей истории
branch     Вывод списка, создание или удаление веток
commit     Запись изменений в репозиторий
merge      Объединение одной или нескольких историй разработки вместе
rebase     Повторное применение коммитов над вершущей другой ветки
reset      Сброс текущего состояния HEAD на указанное состояние
switch     Switch branches
tag        Создание, вывод списка, удаление или проверка метки, подписанной с помощью GPG

совместная работа (смотрите также: git help workflows)
fetch      Загрузка объектов и ссылок из другого репозитория
pull       Извлечение изменений и объединение с другим репозиторием или локальной веткой
push       Обновление внешних ссылок и связанных объектов

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.
farida_gulieva@farida-gulieva-VirtualBox:~$
```

Figure 2.1: Загрузка пакетов

Зададим имя и email владельца репозитория, кодировку и прочие параметры.



```
farida_gulieva@farida-gulieva-VirtualBox:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/farida_gulieva/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/farida_gulieva/.ssh/id_ed25519
Your public key has been saved in /home/farida_gulieva/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:asEpWWSFoe8tIzcjYAKII4JRnNv77d8UNPmbapoF/I farida_gulieva@farida-gulieva-VirtualBox
The key's randomart image is:
+--[ED25519 256]--+
|
|=. 00.0..|
|*+. 0+. = 0 0|
|0. .+* = 0 0|
|oo=. 0 0 0|
|. 0.S . + .|
|.. 0 . . 0|
|.. + . 0|
|.. 0.E .|
|..+0..|
+-----[SHA256]-----+
farida_gulieva@farida-gulieva-VirtualBox:~$
```

Figure 2.4: ed25519

## Создаем GPG ключ

```
farida_gulieva@farida-gulieva-VirtualBox: ~
Файл Правка Вид Поиск Терминал Справка
Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? 0
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печатать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печатать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
gpg: /home/farida_gulieva/.gnupg/trustdb.gpg: создана таблица доверия
gpg: ключ 43C73CE4DF80171D помечен как абсолютно доверенный
gpg: создан каталог '/home/farida_gulieva/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/home/farida_gulieva/.gnupg/openpgp-revocs.d/AEF7A261C0E6B22C208237F843C73CE4DF8017
ID.rev'.
открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2022-09-05 [SC]
      AEF7A261C0E6B22C208237F843C73CE4DF80171D
uid     faridagulieva <1032215201@pfur.ru>
sub     rsa4096 2022-09-05 [E]

farida_gulieva@farida-gulieva-VirtualBox:~$ 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/farida_gulieva/.gnupg/pubring.kbx
-----
sec   rsa4096/43C73CE4DF80171D 2022-09-05 [SC]
      AEF7A261C0E6B22C208237F843C73CE4DF80171D
uid     [ абсолютно ] faridagulieva <1032215201@pfur.ru>
ssb     rsa4096/F801F76EC7B1A395 2022-09-05 [E]

farida_gulieva@farida-gulieva-VirtualBox:~$
```

Figure 2.5: GPG ключ

## Добавляем GPG ключ в аккаунт

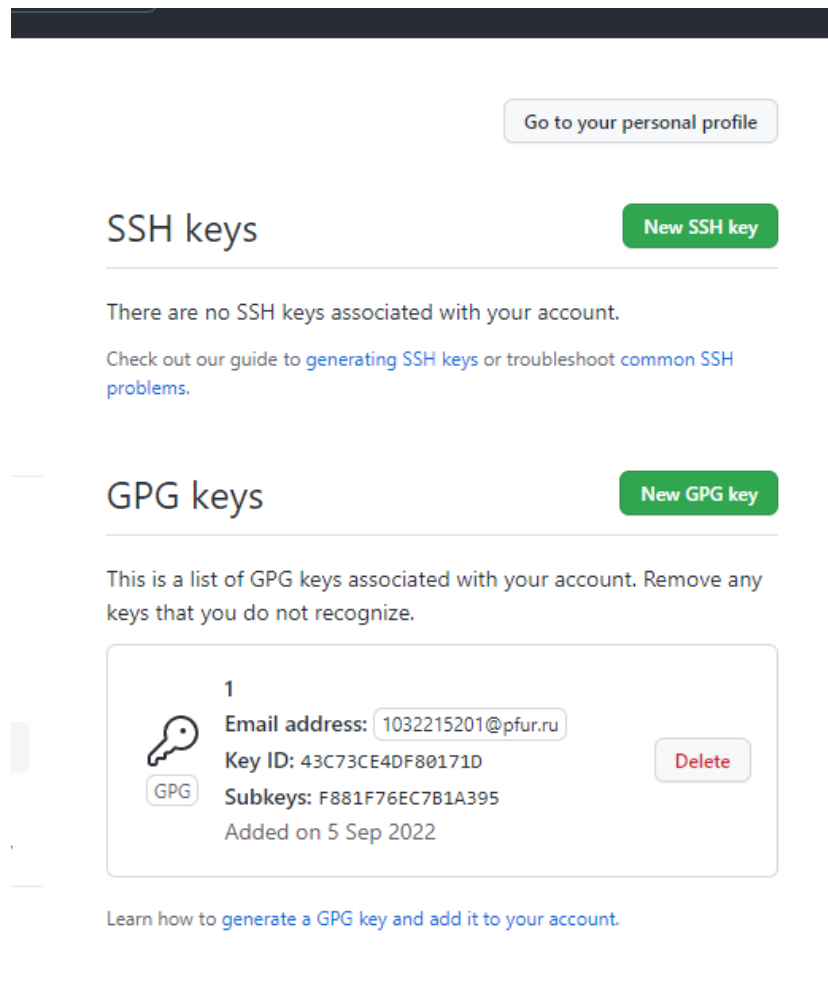


Figure 2.6: GPG ключ

## Настройка автоматических подписей коммитов git

```

108JF02Ks10DBX/1B33DAH/25Ka+Hscce/74haq86uTseyU1S10YguEN/fHpwY4
NYt2q/6r10PNO1M7ex1o7J6aNC1Vyz6e+7yuvf7Wtrodh88ZLx/8fd4GY/v6nh
H1YCZQ8y9n98mF6MfdmpdThap0HfcsF6fU+1oe+UMexVc9nFaJMTja0JELoz/
4btEONz6G900CwqYfoxcfb3KLqT1Cnz0N+11rp2X4Yumb3Bwk39q3dkIDWUJUH1s
CLOafdIVT/HYMDHoCezDf6GfHokCRfIK2LBRm0sD1IKF6KKqC93KA7R5/+pKEA==
=KCaM
-----END PGP PUBLIC KEY BLOCK-----
farida_gulieva@farida-gulieva-VirtualBox:~$
farida_gulieva@farida-gulieva-VirtualBox:~$
farida_gulieva@farida-gulieva-VirtualBox:~$ git config --global user.signingkey 43C73CE4DF80171D
farida_gulieva@farida-gulieva-VirtualBox:~$ git config --global commit.gpgsign true
farida_gulieva@farida-gulieva-VirtualBox:~$ git config --global gpg.program $(which gpg2)
farida_gulieva@farida-gulieva-VirtualBox:~$

```

Figure 2.7: Параметры репозитория

## Настройка gh



```
farida_gulieva@farida-gulieva-VirtualBox:~$
farida_gulieva@farida-gulieva-VirtualBox:~$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? SSH
? Upload your SSH public key to your GitHub account? /home/farida_gulieva/.ssh/id_rsa.pub
? How would you like to authenticate GitHub CLI? Login with a web browser

First copy your one-time code: 3991-FFES
- Press Enter to open github.com in your browser...
- Authentication complete. Press Enter to continue...
2022-09-05T11:31:05Z ERROR viaduct::backend::ffi Missing HTTP status
2022-09-05T11:31:05Z ERROR viaduct::backend::ffi Missing HTTP status

- gh config set -h github.com git_protocol ssh
Configured git protocol
Uploaded the SSH key to your GitHub account: /home/farida_gulieva/.ssh/id_rsa.pub
Logged in as faridagulieva
farida_gulieva@farida-gulieva-VirtualBox:~$
```

Figure 2.8: Связь репозитория с аккаунтом

## Загрузка шаблона репозитория и синхронизация

```
farida_gulieva@farida-gulieva-VirtualBox: ~/work/study/2021-2022/Операционные системы/os-intro
Файл Правка Вид Поиск Терминал Справка
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhp2ZisF/zLDA0zPMSVHdKr4UvC0gu.
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: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 24 (delta 0), reused 18 (delta 0), pack-reused 0
Получение объектов: 100% (24/24), 15.63 Киб | 163.00 Киб/с, готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) записан
трирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован п
о пути «template/report»
Клонирование в «/home/farida_gulieva/work/study/2021-2022/Операционные системы/os-intro/template/presentation»...
remote: Enumerating objects: 71, done.
remote: Counting objects: 100% (71/71), done.
remote: Compressing objects: 100% (49/49), done.
remote: Total 71 (delta 23), reused 68 (delta 20), pack-reused 0
Получение объектов: 100% (71/71), 88.89 Киб | 2.22 Миб/с, готово.
Определение изменений: 100% (23/23), готово.
Клонирование в «/home/farida_gulieva/work/study/2021-2022/Операционные системы/os-intro/template/report»...
remote: Enumerating objects: 78, done.
remote: Counting objects: 100% (78/78), done.
remote: Compressing objects: 100% (52/52), done.
remote: Total 78 (delta 31), reused 69 (delta 22), pack-reused 0
Получение объектов: 100% (78/78), 292.27 Киб | 984.00 Киб/с, готово.
Определение изменений: 100% (31/31), готово.
Submodule path 'template/presentation': checked out '2703b47423792d472694aaf7555a5626dce51a25'
Submodule path 'template/report': checked out 'df7b2ef80f8def3b9a496f8695277469a1a7842a'
farida_gulieva@farida-gulieva-VirtualBox:~/work/study/2021-2022/Операционные системы$ cd ~/work/study/2021-2022/"Опеп
ационные системы"/os-intro
farida_gulieva@farida-gulieva-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ rm package.json
farida_gulieva@farida-gulieva-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ make COURSE=os-intro
farida_gulieva@farida-gulieva-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$
```

Figure 2.9: Загрузка шаблона

## Подготовка репозитория и коммит изменений

```
farida_gulieva@farida-gulieva-VirtualBox: ~/work/study/2021-2022/Операционные системы/os-intro$
create mode 100644 project-personal/stage4/report/Makefile
create mode 100644 project-personal/stage4/report/bib/cite.bib
create mode 100644 project-personal/stage4/report/image/placeimg_800_600_tech.jpg
create mode 100644 project-personal/stage4/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 project-personal/stage4/report/report.md
create mode 100644 project-personal/stage5/presentation/Makefile
create mode 100644 project-personal/stage5/presentation/image/kulyabov.jpg
create mode 100644 project-personal/stage5/presentation/presentation.md
create mode 100644 project-personal/stage5/report/Makefile
create mode 100644 project-personal/stage5/report/bib/cite.bib
create mode 100644 project-personal/stage5/report/image/placeimg_800_600_tech.jpg
create mode 100644 project-personal/stage5/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 project-personal/stage5/report/report.md
create mode 100644 project-personal/stage6/presentation/Makefile
create mode 100644 project-personal/stage6/presentation/image/kulyabov.jpg
create mode 100644 project-personal/stage6/presentation/presentation.md
create mode 100644 project-personal/stage6/report/Makefile
create mode 100644 project-personal/stage6/report/bib/cite.bib
create mode 100644 project-personal/stage6/report/image/placeimg_800_600_tech.jpg
create mode 100644 project-personal/stage6/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 project-personal/stage6/report/report.md
farida_gulieva@farida-gulieva-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$
farida_gulieva@farida-gulieva-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$
farida_gulieva@farida-gulieva-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$
farida_gulieva@farida-gulieva-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ git push
Перечисление объектов: 21, готово.
Подсчет объектов: 100% (21/21), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (17/17), готово.
Запись объектов: 100% (20/20), 311.68 КиБ | 2.43 Миб/с, готово.
Всего 20 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:faridagulieva/study-2021-2022_os-intro.git
796ac48..08f5c10 master -> master
farida_gulieva@farida-gulieva-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$
```

Figure 2.10: Первый коммит

## **3 Вывод**

Мы приобрели практические навыки работы с сервисом github.

## 4 Контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?

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. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

- хранилище - пространство на накопителе где расположен репозиторий
- commit - сохранение состояния хранилища
- история - список изменений хранилища (коммитов)
- рабочая копия - локальная копия сетевого репозитория, в которой работает программист. Текущее состояние файлов проекта, основанное на версии, загруженной из хранилища (обычно на последней)

3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

Централизованные системы контроля версий представляют собой приложения типа клиент-сервер, когда репозиторий проекта существует в единственном экземпляре и хранится на сервере. Доступ к нему осуществлялся через специальное клиентское приложение. В качестве примеров таких программных продуктов можно привести CVS, Subversion.

Распределенные системы контроля версий (Distributed Version Control System, DVCS) позволяют хранить репозиторий (его копию) у каждого разработчика, работающего с данной системой. При этом можно выделить центральный репозиторий (условно), в который будут отправляться изменения из локальных и, с ним же эти локальные репозитории будут синхронизироваться. При работе с такой системой, пользователи периодически синхронизируют свои локальные репозитории с центральным и работают непосредственно со своей локальной копией. После внесения достаточного количества изменений в локальную копию они (изменения) отправляются на сервер. При этом сервер, чаще всего, выбирается условно, т.к. в большинстве DVCS нет такого понятия как “выделенный сервер с центральным репозиторием”.

#### 4. Опишите действия с VCS при единоличной работе с хранилищем.

Один пользователь работает над проектом и по мере необходимости делает коммиты, сохраняя определенные этапы.

#### 5. Опишите порядок работы с общим хранилищем VCS.

Несколько пользователей работают каждый над своей частью проекта. При этом каждый должен работать в своей ветки. При завершении работы ветка пользователя сливается с основной веткой проекта.

#### 6. Каковы основные задачи, решаемые инструментальным средством git?

- Ведение истории версий проекта: журнал (log), метки (tags), ветвления (branches).

- Работа с изменениями: выявление (diff), слияние (patch, merge).
- Обеспечение совместной работы: получение версии с сервера, загрузка обновлений на сервер.

7. Назовите и дайте краткую характеристику командам git.

- git config - установка параметров
- git status - полный список изменений файлов, ожидающих коммита
- git add . - сделать все измененные файлы готовыми для коммита.
- git commit -m "[descriptive message]" - записать изменения с заданным сообщением.
- git branch - список всех локальных веток в текущей директории.
- git checkout [branch-name] - переключиться на указанную ветку и обновить рабочую директорию.
- git merge [branch] — соединить изменения в текущей ветке с изменениями из заданной.
- git push - запустить текущую ветку в удаленную ветку.
- git pull - загрузить историю и изменения удаленной ветки и произвести слияние с текущей веткой.

8. Приведите примеры использования при работе с локальным и удалённым репозиториями.

- git remote add [имя] [url] — добавляет удалённый репозиторий с заданным именем;
- git remote remove [имя] — удаляет удалённый репозиторий с заданным именем;
- git remote rename [старое имя] [новое имя] — переименовывает удалённый репозиторий;
- git remote set-url [имя] [url] — присваивает репозиторию с именем новый адрес;

- `git remote show [имя]` — показывает информацию о репозитории.

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

Ветвление — это возможность работать над разными версиями проекта: вместо одного списка с упорядоченными коммитами история будет расходиться в определённых точках. Каждая ветвь содержит легковесный указатель HEAD на последний коммит, что позволяет без лишних затрат создать много веток. Ветка по умолчанию называется `master`, но лучше назвать её в соответствии с разрабатываемой в ней функциональностью.

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

Зачастую нам не нужно, чтобы Git отслеживал все файлы в репозитории, потому что в их число могут входить:

# Список литературы

1. Лекция Системы контроля версий
2. GitHub для начинающих