

# **Отчёт по лабораторной работе №2**

**Управление версиями**

Александра Александрова

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Вывод</b>	<b>17</b>
<b>4</b>	<b>Контрольные вопросы</b>	<b>18</b>

# Список иллюстраций

2.1	Загрузка пакетов . . . . .	7
2.2	Параметры репозитория . . . . .	8
2.3	rsa-4096 . . . . .	9
2.4	ed25519 . . . . .	10
2.5	GPG ключ . . . . .	11
2.6	GPG ключ . . . . .	12
2.7	Параметры репозитория . . . . .	13
2.8	Связь репозитория с аккаунтом . . . . .	14
2.9	Загрузка шаблона . . . . .	15
2.10	Первый коммит . . . . .	16

## **Список таблиц**

# 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.

```
aabrazhko@aabrazhko:~$ git
использование: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
                [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
                [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--no-lazy-fetch]
                [--no-optional-locks] [--no-advice] [--bare] [--git-dir=<path>]
                [--work-tree=<path>] [--namespace=<name>] [--config-env=<name>=<envvar>]
                <command> [<args>]
```

Стандартные команды Git используемые в различных ситуациях:

создание рабочей области (смотрите также: `git help tutorial`)

<code>clone</code>	Клонирование репозитория в новый каталог
<code>init</code>	Создание пустого репозитория Git или переинициализация существующего

работа с текущими изменениями (смотрите также: `git help everyday`)

<code>add</code>	Добавление содержимого файла в индекс
<code>mv</code>	Перемещение или переименование файла, каталога или символической ссылки
<code>restore</code>	Восстановление файлов в рабочем каталоге
<code>rm</code>	Удаление файлов из рабочего каталога и индекса

Рис. 2.1: Загрузка пакетов

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

```
aabrazhko@aabrazhko:~$ git config --global user.name "aabrazhko"
aabrazhko@aabrazhko:~$ git config --global user.email "1132246887@yandex.ru"
aabrazhko@aabrazhko:~$ git config --global core.quotepath false
aabrazhko@aabrazhko:~$ git config --global init.defaultBranch master
aabrazhko@aabrazhko:~$ git config --global core.autocrlf input
aabrazhko@aabrazhko:~$ git config --global core.safecrlf warn
aabrazhko@aabrazhko:~$
```

Рис. 2.2: Параметры репозитория

Создаем SSH ключи



```

aabrazhko@aabrazhko:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/aabrazhko/.ssh/id_rsa):
Created directory '/home/aabrazhko/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aabrazhko/.ssh/id_rsa
Your public key has been saved in /home/aabrazhko/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:TVeMDlHecd03diT2eZEqBmyANUHLmcNPC05nFs/79Fg aabrazhko@aabrazhko
The key's randomart image is:
+---[RSA 4096]-----+
|      +B=..0.00.+ |
|      .. +=..0+=+  |
|      ..0o+0.0+=+  |
|      . oB+.. ++   |
|      .S+o+ . . o  |
|      +   .       |
|      . . E       |
|      o +        |
|      o .        |
+----[SHA256]-----+
aabrazhko@aabrazhko:~$

```

Рис. 2.3: rsa-4096

```

aabrazhko@aabrazhko:~$
aabrazhko@aabrazhko:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/aabrazhko/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aabrazhko/.ssh/id_ed25519
Your public key has been saved in /home/aabrazhko/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:5fdTKHdt1I6ZcSret6BKF7M2mMXk9RKuAkNMLrt2Xww aabrazhko@aabrazhko
The key's randomart image is:
+--[ED25519 256]--+
|      0.      |
|      +.      |
|      0. .. 0. +|
|      .. 0+Eo 0@.|
|      oS .**oB.*|
|      oo.+o=0.+ |
|      . .=. *0.+ |
|      . +.0 .0. |
|      ... .    |
+----[SHA256]-----+
aabrazhko@aabrazhko:~$

```

Рис. 2.4: ed25519

Создаем GPG ключ

```
aabrazhko@aabrazhko:~  
  
<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: aabrazhko  
Адрес электронной почты: 1132246807@rudn.university  
Примечание:  
Вы выбрали следующий идентификатор пользователя:  
"aabrazhko <1132246807@rudn.university>"  
  
Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? O  
Необходимо получить много случайных чисел. Желательно, чтобы Вы  
в процессе генерации выполняли какие-то другие действия (печать  
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 (печать  
на клавиатуре, движения мыши, обращения к дискам); это даст генератору  
случайных чисел больше возможностей получить достаточное количество энтропии.  
gpg: /home/aabrazhko/.gnupg/trustdb.gpg: создана таблица доверия  
gpg: создан каталог '/home/aabrazhko/.gnupg/openpgp-revocs.d'  
gpg: сертификат отзыва записан в '/home/aabrazhko/.gnupg/openpgp-revocs.d/6EDA64AEC28C8EB297801949C2EFD  
C69F432DD5.rev'.  
  
открытый и секретный ключи созданы и подписаны.  
  
pub  rsa4096 2025-02-19 [SC]  
      6EDA64AEC28C8EB297801949C2EFD C69F432DD5  
uid          aabrazhko <1132246807@rudn.university>  
sub  rsa4096 2025-02-19 [E]  
  
aabrazhko@aabrazhko:~$
```

Рис. 2.5: GPG ключ

Добавляем GPG ключ в аккаунт

```

aabrazhko@aabrazhko:~$ 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
[keyboxd]
-----
sec   rsa4096/C2EFD69F432DD5 2025-02-19 [SC]
      6EDA64AEC28C8EB297801949C2EFD69F432DD5
uid           [ абсолютно ] aabrazhko <1132246807@rudn.university>
ssb   rsa4096/76A8E63D30D8ADB 2025-02-19 [E]

aabrazhko@aabrazhko:~$ gpg --armor --export C2EFD69F432DD5
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQIN8Ge1p0o8EACbUcoimVSvDSbv+euAn1JUfBxyl4ZLQrmY0WYPNuuAFTWv0cj
Y40glL1fDs9yJFQof9tigX48FIZTyVx8qRq6YCc2IShmDhjBMSkl+Xi6F8u+/OCr
NSFstfSb0KP6i3patwc110sgooYCAU7d/xQ/Zb0g272DW4fBJ1Ej/BIX0d8I6TU6
iWINGuNSQK1PI7JTzr1rByjM9aYkq2WVm2yIBMXzWAbPt59Mm7I2o7cTOWGMP
85y2x0VhrdviPC32S1QDrE+Ki8XImCMKHPZ7NrbphYRFqCYLueDvLwZEz7L3S8I
HonePJIZPjeXyi04JXoHoNApvi1qL00mGqUXd5RgXE/KBFqHzj7T44P0ERP7ADB
Z+KI8jdWiEfpvBa1BauxaJXNGmhQ6GsQrEfVQMLWKYxFIdqn7ceK8X9IYJiLrB4
3NxoXmnrj3apx8y/VoDlaziJSTnVvWBoVzmjhmKwVd1fZptTv57X+WLxeIKhFrhd
YZH050pt0e9VoAJ+mMIUqzIpAssL9c03NV6G43JURQ9dNOQHj0id10ny4e3uF9KQ
fuvzs0Zl0B9RyUpK0mczHW+6pU8P+IfoPpP2541Qa1pHbaGnNcksKcgDFZ0+2FLZ
/6x/jlIWMRY0dW0MnNNviUn96twec6pp3JNQXYei0TIqtFDFjFIgNUR9zQARAQAB
tCZhYVYyYXpoa28gPDExMzIyNDY4MDdAcnVkb151bm12ZXJzaXR5PokCUQQAQgA
OxYhBG7aZK7CjI6yl4AZScLv3Gn0Mt3VBQJntadKAhsDBQsJCAcCAiICBhUKCQGL
AgQWAgMBAH4HAheAAAJEMLv3Gn0Mt3VG+AP/jV7/Vj+27K950um5sIDArbnC1pI
yg+zPCGa6B+iGfYqbrm6BLSR3CvWg8aAk+GLtyGXP95y8sgZQWYb0Hp2n6Q+X1Yx
eikW3ePvI0dTIWC4i/emRIjTaCE/QITaEjB0ZFFwKZgJz0NJtTnBZ655WhIofPDb
+xEzDDcz+10pDV4v0GiYw5pFAfcSMP33tmB2BPb7V7u7atQqAXhTnbF59ayXIIdBE

```

Рис. 2.6: GPG ключ

Настройка автоматических подписей коммитов git

```
aabrazhko@aabrazhko:~$ git config --global user.signingkey C2EFD069F4320005  
aabrazhko@aabrazhko:~$ git config --global commit.gpgsign true  
aabrazhko@aabrazhko:~$ git config --global gpg.program $(which gpg2)  
aabrazhko@aabrazhko:~$
```

Рис. 2.7: Параметры репозитория

Настройка gh

```
aabrazhko@aabrazhko:~$ gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? SSH
? Upload your SSH public key to your GitHub account? /home/aabrazhko/.ssh/id_rsa.pub
? Title for your SSH key: GitHub CLI
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 9B20-1857
Press Enter to open https://github.com/login/device in your browser...
✓ Authentication complete.
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ Uploaded the SSH key to your GitHub account: /home/aabrazhko/.ssh/id_rsa.pub
✓ Logged in as aabrazhko
aabrazhko@aabrazhko:~$
```

Рис. 2.8: Связь репозитория с аккаунтом

Загрузка шаблона репозитория и синхронизация

```
aabrazhko@aabrazhko:~$ mkdir -p ~/work/study/2024-2025/"Операционные системы"
aabrazhko@aabrazhko:~$ cd ~/work/study/2024-2025/"Операционные системы"
aabrazhko@aabrazhko:~/work/study/2024-2025/Операционные системы$ gh repo create os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository aabrazhko/os-intro on GitHub
https://github.com/aabrazhko/os-intro
aabrazhko@aabrazhko:~/work/study/2024-2025/Операционные системы$ git clone --recursive git@github.com:aabrazhko/os-intro.git os-intro
Клонирование в «os-intro»...
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TjDhnpZisf/zLDA0zPM6vHdK4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Рис. 2.9: Загрузка шаблона

## Подготовка репозитория и коммит изменений

```

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 100755 project-personal/stage6/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_fignos.py
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_secnos.py
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/core.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/main.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 project-personal/stage6/report/report.md
aabrazhko@aabrazhko:~/work/study/2024-2025/Операционные системы/os-intro$ git push
Перечисление объектов: 38, готово.
Подсчет объектов: 100% (38/38), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (37/37), 342.27 Киб | 2.24 Миб/с, готово.
Total 37 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:aabrazhko/os-intro.git
  6cf07e7..96e0b47  master -> master
aabrazhko@aabrazhko:~/work/study/2024-2025/Операционные системы/os-intro$

```

Рис. 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: