

# **Отчёт по лабораторной работе 2**

**Первоначальная настройка git**

Сидорова Наталья Андреевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>19</b>
	<b>Список литературы</b>	<b>20</b>

## Список иллюстраций

4.1	Установка ПО . . . . .	8
4.2	Базовая настройка . . . . .	9
4.3	Ключи SSH . . . . .	10
4.4	Ключ pgr . . . . .	11
4.5	Учетная запись . . . . .	11
4.6	Отпечаток ключа . . . . .	12
4.7	Добавление ключа . . . . .	12
4.8	Настройка gh . . . . .	13
4.9	Автоматические подписи . . . . .	14
4.10	Репозиторий курса . . . . .	15
4.11	Настройка каталога . . . . .	15
4.12	Отправка файлов . . . . .	16

## **Список таблиц**

# 1 Цель работы

Изучить идеологию и применение средств контроля версий. Освоить умения по работе с git.

## 2 Задание

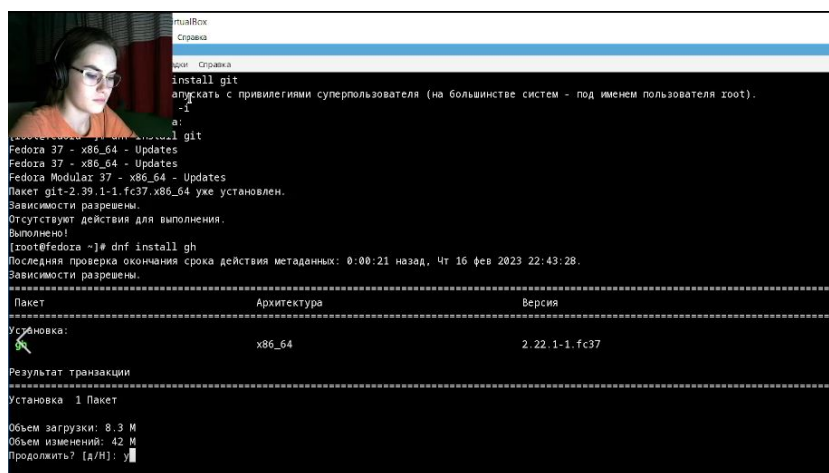
1. Зарегистрироваться на Github
2. Создать базовую конфигурацию для работы с git
3. Создать ключ SSH
4. Создать ключ PGP
5. Настроить подписи git
6. Создать локальный каталог для выполнения заданий по предмету

### **3 Теоретическое введение**

Системы контроля версий применяются при работе нескольких человек над одним проектом. Основное дерево хранится на репозитории. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, размещать изменения от разных участников проекта.

## 4 Выполнение лабораторной работы

Установила git и gh (рис. 4.1).

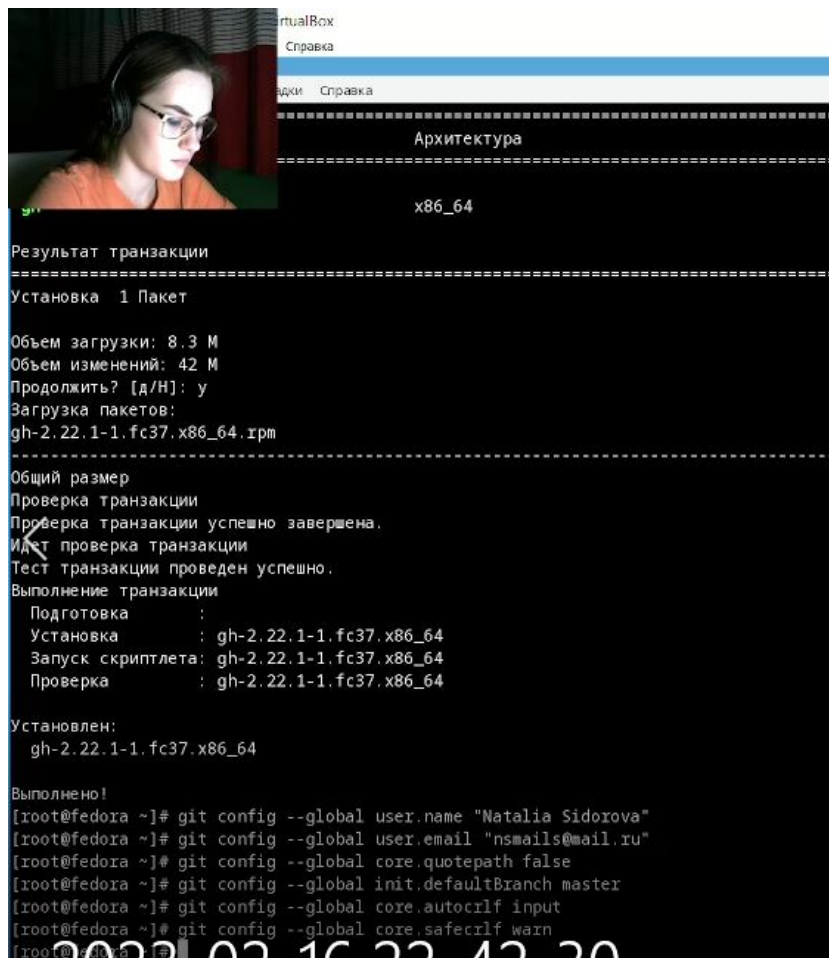


```
root@fedora: ~# dnf install git
install git
аппликация с привилегиями суперпользователя (на большинстве систем - под именем пользователя root).
-1
S:
Установка git
Fedora 37 - x86_64 - Updates
Fedora 37 - x86_64 - Updates
Fedora Modular 37 - x86_64 - Updates
Пакет git-2.39.1-1.fc37.x86_64 уже установлен.
Зависимости разрешены.
Отсутствуют действия для выполнения.
Выполнено!
[root@fedora ~]# dnf install gh
Последняя проверка окончания срока действия метаданных: 0:00:21 назад, Чт 16 фев 2023 22:43:28.
Зависимости разрешены.
=====
Пакет      Архитектура      Версия
-----
Установка:
gh          x86_64            2.22.1-1.fc37
=====
Результат транзакции
=====
Установка 1 Пакет
Объем загрузки: 8.3 М
Объем изменений: 42 М
Продолжить? [д/н]: y
```

Рис. 4.1: Установка ПО

Произвела базовую настройку git - задала имя и email владельца репозитория, настроила utf-8 в выводе сообщений git, задала имя начальной ветки(master), задала параметр autocrlf и параметр safecrlf (рис. 4.2).

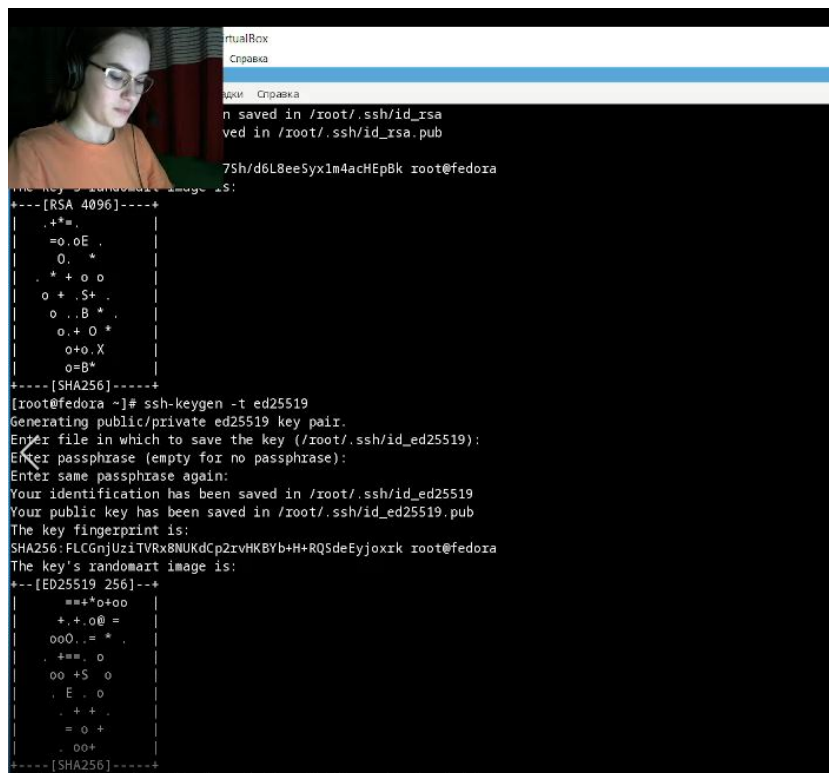




```
virtualBox
Справка
Справка
Архитектура
x86_64
Результат транзакции
=====
Установка 1 Пакет
Объем загрузки: 8.3 М
Объем изменений: 42 М
Продолжить? [д/н]: у
Загрузка пакетов:
gh-2.22.1-1.fc37.x86_64.rpm
-----
Общий размер
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка :
Установка : gh-2.22.1-1.fc37.x86_64
Запуск скрипглета: gh-2.22.1-1.fc37.x86_64
Проверка : gh-2.22.1-1.fc37.x86_64
Установлен:
gh-2.22.1-1.fc37.x86_64
Выполнено!
[root@fedora ~]# git config --global user.name "Natalia Sidorova"
[root@fedora ~]# git config --global user.email "nsmails@mail.ru"
[root@fedora ~]# git config --global core.quotepath false
[root@fedora ~]# git config --global init.defaultBranch master
[root@fedora ~]# git config --global core.autocrlf input
[root@fedora ~]# git config --global core.safecrlf warn
[root@fedora ~]#
```

Рис. 4.2: Базовая настройка

Создала ключи SSH - по алгоритму rsa и по алгоритму ed25519 (рис. 4.3).



```
VirtualBox
Справка
вкл: Справка
n saved in /root/.ssh/id_rsa
ved in /root/.ssh/id_rsa.pub
7Sh/d6L8eeSyx1m4acHEpBk root@fedora
The key's fingerprint is:
SHA256: 7Sh/d6L8eeSyx1m4acHEpBk
The key's randomart image is:
+---[RSA 4096]-----+
|
|  .+*+
|  =0.oE .
|  0. *
|  .+ + o o
|  o + .S+ .
|  o..B *
|  o.+ O *
|  o+o.X
|  o=B*
+-----[SHA256]-----+
[root@fedora ~]# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256: FLCGnjUziTVRx8NUKdCp2rvHKBYb+H+RQsdeEyjoxrk root@fedora
The key's randomart image is:
+---[ED25519 256]---+
|
|  ==+*o+oD
|  +,+o@ =
|  ooO..= *
|  .+==, o
|  oo+S o
|  E . o
|  .+ + .
|  = o +
|  .oo+
+-----[SHA256]-----+
```

Рис. 4.3: Ключи SSH

Создала ключ pgr (рис. 4.4).



```
[root@fedora ~]# gpg --full-generate-key
gpg (GnuPG) 2.3.8; 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 '/root/.gnupg'
gpg: создан щит с ключами '/root/.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
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
  <p> = срок действия ключа - п дней
  <p>w = срок действия ключа - п недель
  <p>m = срок действия ключа - п месяцев
  <p>y = срок действия ключа - п 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: Natalia
Адрес электронной почты: nsmails@mail.ru
```

Рис. 4.4: Ключ ргр

Учетная запись на Github уже была создана (рис. 4.5).

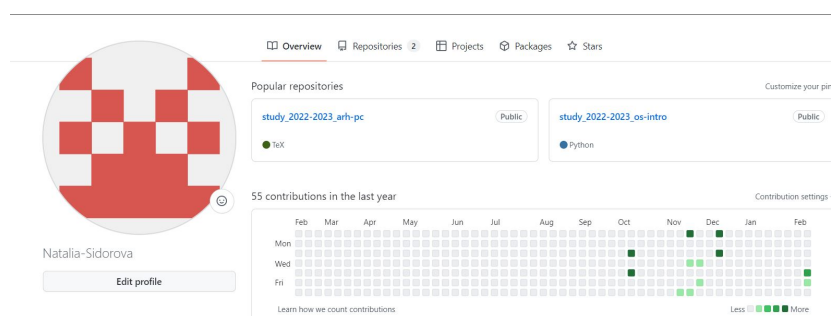


Рис. 4.5: Учетная запись

Скопировала отпечаток приватного ключа (рис. 4.6).

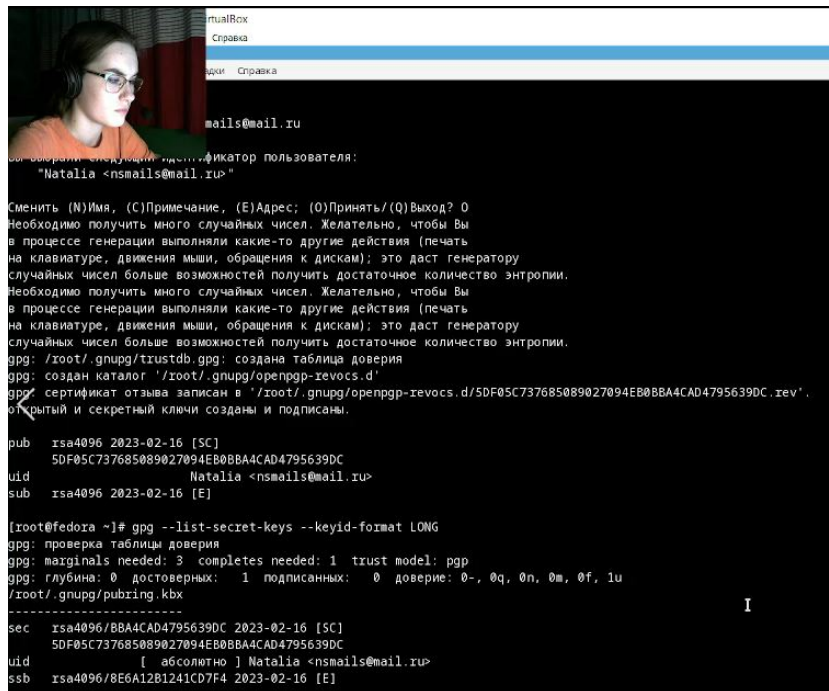


Рис. 4.6: Отпечаток ключа

Добавила скопированный ключ в Github (рис. 4.7).

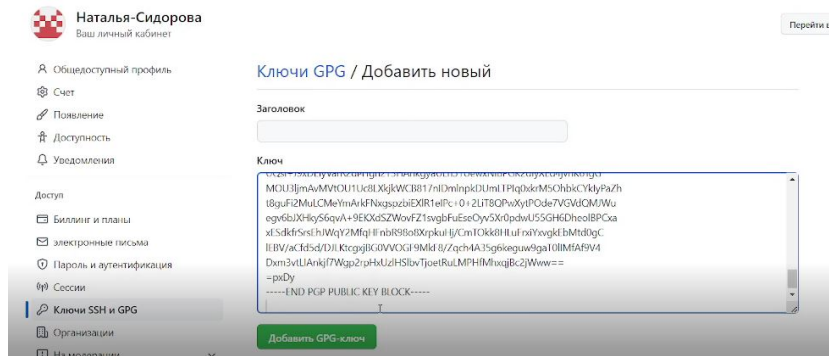



Рис. 4.7: Добавление ключа

Настроила gh (рис. 4.8).

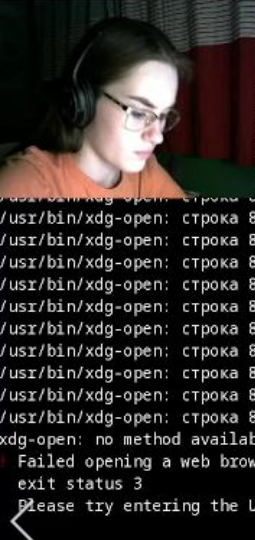


```
а regular user's session is not supported. ($XAUTHORITY is /run/
81: x-www-browser: команда не найдена
not get canonical path for /root/.mozilla/firefox/* restorecon: N
а regular user's session is not supported. ($XAUTHORITY is /run/
81: iceweasel: команда не найдена
/usr/bin/xdg-open: строка 881: seamonkey: команда не найдена
/usr/bin/xdg-open: строка 881: mozilla: команда не найдена
/usr/bin/xdg-open: строка 881: epiphany: команда не найдена
/usr/bin/xdg-open: строка 881: konqueror: команда не найдена
/usr/bin/xdg-open: строка 881: chromium: команда не найдена
/usr/bin/xdg-open: строка 881: chromium-browser: команда не найдена
/usr/bin/xdg-open: строка 881: google-chrome: команда не найдена
/usr/bin/xdg-open: строка 881: www-browser: команда не найдена
/usr/bin/xdg-open: строка 881: links2: команда не найдена
/usr/bin/xdg-open: строка 881: elinks: команда не найдена
/usr/bin/xdg-open: строка 881: links: команда не найдена
/usr/bin/xdg-open: строка 881: lynx: команда не найдена
/usr/bin/xdg-open: строка 881: w3m: команда не найдена
xdg-open: no method available for opening 'https://github.com/login/device'
failed opening a web browser at https://github.com/login/device
exit status 3
Please try entering the URL in your browser manually

[root@fedora ~]# 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? /root/.ssh/id_rsa.pub
? Title for your SSH key: GitHub CLI
? How would you like to authenticate GitHub CLI? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'admin:public_key'.
? Paste your authentication token: *****
- gh config set -h github.com git_protocol ssh
- Configured git protocol
- Uploaded the SSH key to your GitHub account: /root/.ssh/id_rsa.pub
- Logged in as Natalia-Sidorova
[root@fedora ~]#
```

Рис. 4.8: Настройка gh

Настроила автоматические подписи коммитов git (рис. 4.9).



```
VirtualBox
Справка

...
a regular user's session is not supported. ($XAUTHORITY is
81: iceweasel: команда не найдена
81: seamonkey: команда не найдена
81: mozilla: команда не найдена
81: epiphany: команда не найдена
81: konqueror: команда не найдена
81: chromium: команда не найдена
81: chromium-browser: команда не найдена
81: google-chrome: команда не найдена
81: www-browser: команда не найдена
81: links2: команда не найдена
81: elinks: команда не найдена
81: links: команда не найдена
81: lynx: команда не найдена
81: w3m: команда не найдена
xdg-open: no method available for opening 'https://github.com/login/device'
Failed opening a web browser at https://github.com/login/device
exit status 3
Please try entering the URL in your browser manually

[root@fedora ~]# 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? /root/.ssh/id_rsa.pub
? Title for your SSH key: GitHub CLI
? How would you like to authenticate GitHub CLI? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'admin:public_key'.
? Paste your authentication token: *****
- gh config set -h github.com git_protocol ssh
- Configured git protocol
- Uploaded the SSH key to your GitHub account: /root/.ssh/id_rsa.pub
- Logged in as Natalia-Sidorova
[root@fedora ~]# git config --global user.signingkey BBA4CAD4795639DC
[root@fedora ~]# git config --global commit.gpgsign true
[root@fedora ~]# git config --global gpg.program $(which gpg2)
```

Рис. 4.9: Автоматические подписи

Создала репозиторий курса на основе шаблона (рис. 4.10).





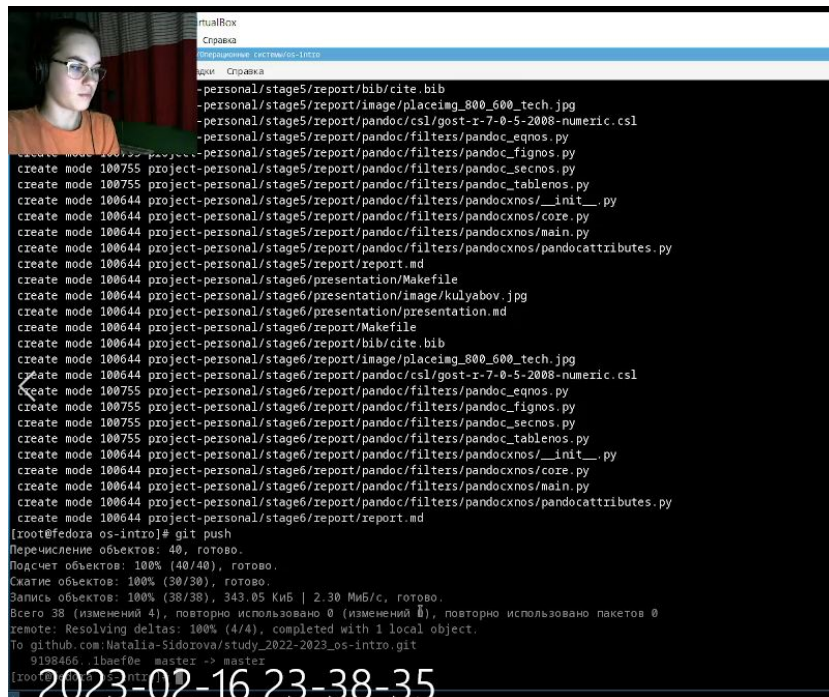


Рис. 4.12: Отправка файлов

Ответы на контрольные вопросы:

Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются? Система контроля версий - ПО для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям. Системы контроля версий применяются для хранения полной истории изменений, причин всех производимых изменений, откат изменений, поиск причины ошибок в программе, совместная работа группы над проектом.

Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия. Репозиторий - хранилище версий. В нем хранятся все документы с историей их изменения, а также другая служебная информация. Commit - отслеживание изменений, сохраняет разницу в изменениях. Рабочая копия - копия проекта, связанная с репозиторием (текущее состояние файлов



проекта) История - хранит все изменения в проекте и позволяет обратиться к необходимым данным.

Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида. Централизованные VCS - Subversion, CVS, TFS, VAULT, AccuRev: одно основное хранилище всего проекта, каждый пользователь копирует себе необходимые файлы, изменяет их и добавляет эти изменения обратно. Децентрализованные VCS - Git, Mercurial, Bazaar: у каждого пользователя свой вариант репозитория, есть возможность добавлять и забирать изменения из любого репозитория. В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. В центральном репозитории не является обязательным, выполнение большинства функций по управлению версиями осуществляется специальным сервером.

Опишите действия с VCS при единоличной работе с хранилищем. Создаем и подключаем удаленный репозиторий. Затем по мере изменения проекта отправляем изменения на сервер.

Опишите порядок работы с общим хранилищем VCS. Пользователь перед началом работы получает нужную ему версию файлов. После внесения изменений пользователь размещает новую версию в хранилище. Предыдущие версии не удаляются.

Каковы основные задачи, решаемые инструментальным средством git? Хранить информацию о всех изменениях в коде и обеспечение удобства командной работы над кодом.

Назовите и дайте краткую характеристику командам git. `git init` создание нового дерева репозитория `git pull` получение изменений текущего дерева из центрального репозитория `git push` - отправка всех произведенных изменений локального дерева в центральный репозиторий `git status` - просмотр списка измененных файлов в текущей директории `git diff` - просмотр текущих изменений `git add` - добавить измененные файлы/каталоги `git rm` - удаление файла/ката-

лога из индекса репозитория `git commit -am` 'Описание коммита' - сохранить все добавленные изменения и все измененные файлы `git checkout -b` имя ветки - создание новой ветки, базирующейся на текущей `git push origin` имя ветки - отправка изменений конкретной ветки в центральный репозиторий `git merge -no-ff` имя ветки - слияние ветки с текущим деревом `git branch -d` имя ветки - удаление локальной уже слитой с основным деревом ветки `git branch -D` имя ветки - принудительное удаления локальной ветки `git push origin:имя ветки` - удаление ветки с центрального репозитория

Приведите примеры использования при работе с локальным и удалённым репозиториями. `git push -all(push origin master/branch)`

Что такое и зачем могут быть нужны ветви (branches)? Ветвление - один из параллельных участков истории в одном хранилище, исходящих из одной версии. Обычно есть главная ветка (ствол), между ветками возможно слияние. Используются для разработки новых функций.

Как и зачем можно игнорировать некоторые файлы при commit? Во время работы над проектом могут создаваться файлы, которые не требуется добавлять в репозиторий. Можно прописать шаблоны игнорируемых при добавлении в репозиторий типов файлов в файл `.gitignore`.

## 5 Выводы

В ходе проделанной работы я изучила средства контроля версий и научилась применять их, освоила работу с git, научилась подключать репозитории, добавлять и удалять файлы.

## Список литературы

1. О системе контроля версий [Электронный ресурс]. 2016. URL: <https://git-scm.com/book/ru/v2/Введение-О-системе-контроля-версий> .
2. Системы контроля версий [Электронный ресурс]. 2016. URL: <http://uii.mpei.ru/study/courses> . ::: {#refs} :::