

# **Отчёт по лабораторной работе № 2**

**НММбд-02-22**

Никулина Ксения Ильинична

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
3.1	Установка программного обеспечения . . . . .	6
3.2	Базовая настройка git . . . . .	7
3.3	Создали ключи ssh . . . . .	8
3.4	Создали ключи pgr . . . . .	9
3.5	Настройка github . . . . .	10
3.6	Добавление PGP ключа в GitHub . . . . .	11
3.7	Настройка автоматических подписей коммитов git . . . . .	13
3.8	Настройка gh . . . . .	13
3.9	Создание репозитория курса на основе шаблона . . . . .	14
3.10	Настройка каталога курса . . . . .	14
<b>4</b>	<b>Выводы</b>	<b>16</b>
<b>5</b>	<b>Ответы на контрольные вопросы</b>	<b>17</b>

# Список иллюстраций

3.1	6
3.2	6
3.3	7
3.4	7
3.5	7
3.6	7
3.7	7
3.8	8
3.9	8
3.10	9
3.11	10
3.12	10
3.13	11
3.14	11
3.15	12
3.16	12
3.17	13
3.18	13
3.19	13
3.20	14
3.21	14
3.22	14
3.23	15
3.24	15
3.25	15
3.26	15
3.27	15
5.1	18
5.2	19
5.3	20

# 1 Цель работы

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

## 2 Задание

- Установить и настроить ПО для работы с git.

## 3 Выполнение лабораторной работы

### 3.1 Установка программного обеспечения

Установили git:(рис. [3.1])

```
knikulina@fedora ~]$ sudo -i
[sudo] пароль для knikulina:
[root@fedora ~]# dnf install git
Последняя проверка окончания срока действия метаданных: 0:23:13 назад, Сб
 2023 12:04:18.
Пакет git-2.35.1-1.fc36.x86_64 уже установлен.
Зависимости разрешены.
Отсутствуют действия для выполнения.
Выполнено!
```

Рис. 3.1: .

Установили gh:(рис. [3.2])

```
[root@fedora ~]# dnf install gh
Последняя проверка окончания срока действия метаданных: 0:23:48 назад, Сб 18 фев
 2023 12:04:18.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий  Размер
=====
Установка:
gh          x86_64       2.23.0-1.fc36  updates      8.2 М
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 8.2 М
Объем изменений: 41 М
Продолжить? [д/н]: д
Загрузка пакетов:
gh-2.23.0-1.fc36.x86_64.rpm          3.0 МВ/с | 8.2 МВ    00:02
-----
Общий размер          2.3 МВ/с | 8.2 МВ    00:03
Проверка транзакции
```

Рис. 3.2: .

## 3.2 Базовая настройка git

Задали имя и email владельца репозитория: (рис. [3.3])

```
banon@fedora: ~$ git config --global user.name "kinikulina"
[banon@fedora ~]$ git config --global user.email "ksenianik2108@gmail.com"
[banon@fedora ~]$
```

Рис. 3.3: .

Настроили utf-8 в выводе сообщений git:(рис. [3.4])

```
[root@fedora ~]# git config --global core.quotepath false
[root@fedora ~]#
```

Рис. 3.4: .

Настроили верификацию и подписание коммитов git. Задали имя начальной ветки (будем называть её master).(рис. [3.5])

```
[root@fedora ~]# git config --global init.defaultBranch master
[root@fedora ~]#
```

Рис. 3.5: .

Параметр autocrlf:(рис. [3.6])

```
[root@fedora ~]# git config --global core.autocrlf input
[root@fedora ~]#
```

Рис. 3.6: .

Параметр safecrlf: (рис. [3.7])

```
[root@fedora ~]# git config --global core.safecrlf warn
[root@fedora ~]#
```

Рис. 3.7: .

### 3.3 Создали ключи ssh

по алгоритму rsa с ключём размером 4096 бит: (рис. [3.8])

```
[root@fedora ~]# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:+axpj530R1PL24Y0+bHNgANgYSkJ0BmCIRK+sHV4Wic root@fedora
The key's randomart image is:
+---[RSA 4096]-----+
|Bo.+oo...o.      |
|=.++E.o...       |
|..+.o .o         |
|.o.o .o .        |
|..      S . . . .|
|          o .ooo  |
|          + =.ooo  |
|          .* o +oBo|
|          .+. .o+.o|
+-----[SHA256]-----+
[root@fedora ~]#
```

Рис. 3.8: .

по алгоритму ed25519: (рис. [3.9])

```
[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:uWtM603qjvFXfedIKwU54gQY9UhRwk4Y8ZPHSBe4iLM root@fedora
The key's randomart image is:
+---[ED25519 256]---+
|      +0+++      |
|      o+=0       |
|      . +*o+ .   |
|      o . o+o +   |
|      o .So o o   |
|      E . ..o . + |
|      .. +.. + =  |
|      +. =. . o . |
|      .o==0 .     |
+-----[SHA256]-----+
[root@fedora ~]#
```

Рис. 3.9: .



## 3.4 Создали ключи ргр

Сгенерировали ключ (рис. [3.10])

Из предложенных опций выбирали: тип RSA and RSA; размер 4096; выбрали срок действия; значение по умолчанию — 0 (срок действия не истекает никогда). GPG запросил личную информацию, которая сохранится в ключе: Имя. Адрес электронной почты. При вводе email убедились, что он соответствует адресу, используемому на GitHub. (рис. [3.11])

```
[root@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.

gpg: создан каталог '/root/.gnupg'
gpg: создан шит 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
  <n> = срок действия ключа - n дней
  <n>w = срок действия ключа - n недель
  <n>m = срок действия ключа - n месяцев
```

Рис. 3.10: .

```
Примечание: n
Вы выбрали следующий идентификатор пользователя:
"Ksenia Nikulina (n) <ksenianik2108@gmail.com>"

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: /root/.gnupg/trustdb.gpg: создана таблица доверия
gpg: создан каталог '/root/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/root/.gnupg/openpgp-revocs.d/D4D3E35F32F1402
950768C93D552F31523208B.rev'.
открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2023-02-18 [SC]
       D4D3E35F32F140260950768C93D552F31523208B
uid           Ksenia Nikulina (n) <ksenianik2108@gmail.com>
sub   rsa4096 2023-02-18 [E]
```

Рис. 3.11: .

## 3.5 Настройка github

Создайте учётную запись на github.com. (рис. [3.12])

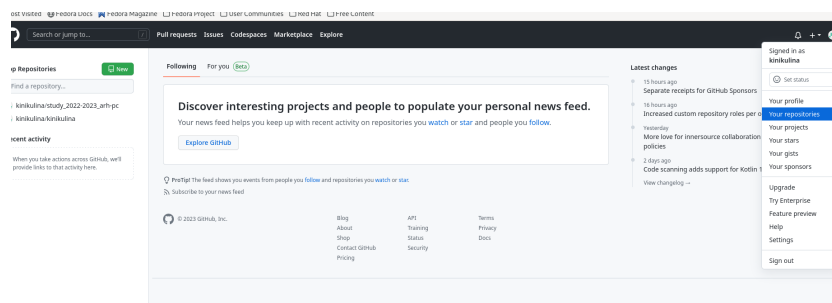


Рис. 3.12: .

Заполните основные данные на github.com. (рис. [3.13])

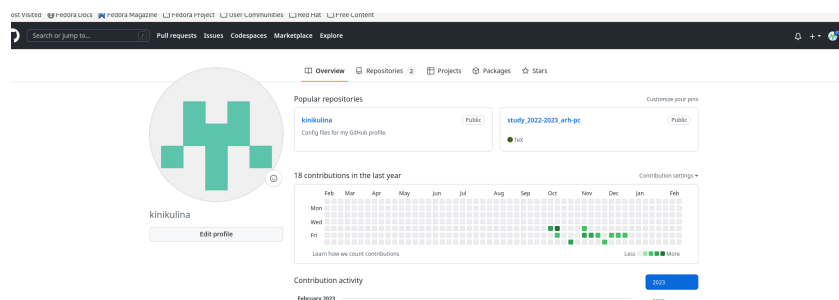


Рис. 3.13: .

## 3.6 Добавление PGP ключа в GitHub

Вывели список ключей и копировали отпечаток приватного ключа: (рис. [3.14])  
Отпечаток ключа — это последовательность байтов, используемая для идентификации более длинного, по сравнению с самим отпечатком ключа.

```
[root@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
/root/.gnupg/pubring.kbx
-----
sec   rsa4096/93D552F31523208B 2023-02-18 [SC]
      D4D3E35F32F140260950768C93D552F31523208B
uid           [ абсолютно ] Ksenia Nikulina (n) <ksenianik2108@gmail.com>
ssb   rsa4096/72A688AC346CAD11 2023-02-18 [E]
```

Рис. 3.14: .

Скопировали сгенерированный PGP ключ в буфер обмена: (рис. [3.15])

```
[root@fedora ~]# gpg --armor --export | 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...
```

Рис. 3.15: .

Перешли в настройки GitHub, нажали на кнопку New GPG key и вставили полученный ключ в поле ввода. (рис. [3.16], [3.17])

GPG keys / Add new

---

Title

Key

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
mQINBFmNDG7vJCvH+iSawGhXRjYQH6vDHCIWQpQz9VeYHvxOgHYNMm2XTBpd2EW
x4l2nmIYB4fnuGnC1BG1TLjkm2c2+trayhOpfDXZ6g9GcdGWzMM8exuZe2Z2IndG
Pcmf4cLGdcVWV9295sbeUreq04CHCwH5D2Fct7egCbf5LQhiai3JGRjC4I7v0pM
HZpDrLPQwFT6Xohk5SrQFyhxhBZh2/WySwgNe3pnTKhLnZ6fsCvoprXSSSTfEe9L
6ZJqbiIoHs2/RIU8s/9J5vaYrk0fqBQuKDwqfnjSHvIDim0n/DIFNjgRIWHoan8ys
6VK4QCHHCAEIs++fA==
=RVx+
```

Рис. 3.16: .

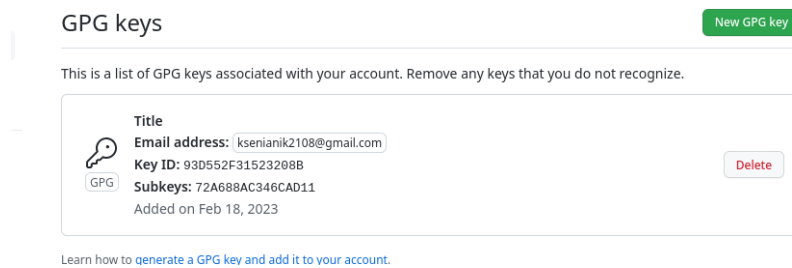


Рис. 3.17: .

## 3.7 Настройка автоматических подписей коммитов git

Используя введённый email, указали Git применять его при подписи коммитов: (рис. [3.18])

```
[root@fedora ~]# git config --global user.signingkey  
[root@fedora ~]# git config --global commit.gpgsign true  
[root@fedora ~]# git config --global gpg.program $(which gpg2)
```

Рис. 3.18: .

## 3.8 Настройка gh

Авторизовались в gh. (рис. [3.19]) Утилита задали несколько наводящих вопросов.

```
[root@fedora ~]# gh auth login  
? What account do you want to log into? GitHub.com  
? What is your preferred protocol for Git operations? HTTPS  
? Authenticate Git with your GitHub credentials? Yes  
? How would you like to authenticate GitHub CLI? Login with a web browser
```

Рис. 3.19: .

## 3.9 Создание репозитория курса на основе шаблона

Создали шаблон рабочего пространства. (рис. [3.20], [3.21], [3.22])

```
[root@fedora ~]# mkdir -p ~/work/study/2022-2023/"Операционные системы"
[root@fedora ~]# cd ~/work/study/2022-2023/"Операционные системы"
[root@fedora Операционные системы]#
```

Рис. 3.20: .

Create a new repository from course-directory-student-template

The new repository will start with the same files and folders as [yamadharm/course-directory-student-template](#).

Owner \* kinikulina / Repository name \* study\_2022-2023\_os-intro ✓

Great repository names study\_2022-2023\_os-intro is available. [piration?](#) How about [psychic-barnacle](#)?

Description (optional)

☒ Public  
Anyone on the internet can see this repository. You choose who can commit.

☐ Private  
You choose who can see and commit to this repository.

☐ Include all branches  
Copy all branches from yamadharm/course-directory-student-template and not just master.

① You are creating a public repository in your personal account.

Create repository from template

Рис. 3.21: .

```
[knikulina@fedora Операционные системы]$ git clone --recursive git@github.com:kinikulina/study_2022-2023_os-intro.git
Клонирование в «study_2022-2023_os-intro»...
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdKr4UvCOqU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

Рис. 3.22: .

## 3.10 Настройка каталога курса

Перешли в каталог курса: (рис. [3.23])

```
[knikulina@fedora Операционные системы]$ cd ~/work/study/2022-2023/"Операционные системы"/study_2022-2023_os-intro  
[knikulina@fedora study_2022-2023_os-intro]$
```

Рис. 3.23: .

Удалили лишние файлы: (рис. [3.24])

```
[knikulina@fedora study_2022-2023_os-intro]$ rm package.json  
[knikulina@fedora study_2022-2023_os-intro]$
```

Рис. 3.24: .

Создали необходимые каталоги: (рис. [3.25])

```
[knikulina@fedora study_2022-2023_os-intro]$ echo os-intro > COURSE  
[knikulina@fedora study_2022-2023_os-intro]$ make
```

Рис. 3.25: .

Отправили файлы на сервер: (рис. [3.26], [3.27])

```
[knikulina@fedora study_2022-2023_os-intro]$ git add .  
[knikulina@fedora study_2022-2023_os-intro]$ git commit -am 'feat(main): make course structure'
```

Рис. 3.26: .

```
[knikulina@fedora study_2022-2023_os-intro]$ git push
```

Рис. 3.27: .

## **4 Выводы**

В ходе выполнения данной лабораторной работы была изучена идеология и применение средств контроля версий и освоены умения по работе с git.



## 5 Ответы на контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются? Система управления версиями (также используется определение «система контроля версий», от англ. Version Control System, VCS или Revision Control System) — программное обеспечение 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, определять, кто и когда сделал то или иное изменение, и многое другое.
2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия. Хранилище (repository), или репозиторий, — место хранения файлов и их версий, служебной информации. Версия (revision), или ревизия, — состояние всего хранилища или отдельных файлов в момент времени («пункт истории»). Commit («трудовой вклад», не переводится) — процесс создания новой версии; иногда синоним версии. Рабочая копия (working copy) — текущее состояние файлов проекта (любой версии), полученных из хранилища и, возможно, измененных.
3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида. Децентрализованные VCS: У каждого пользователя свой вариант (возможно не один) репозитория. Присутствует возможность добавлять и забирать изменения из любого репозитория (Git, Mercurial, Bazaar).

Централизованные VCS : Одно основное хранилище всего проекта Каждый пользователь копирует себе необходимые ему файлы из этого репозитория, изменяет и, затем, добавляет свои изменения обратно (Subversion, CVS, TFS, VAULT, AccuRev)

4. Опишите действия с VCS при единоличной работе с хранилищем. (рис. [5.1])

## Единоличная работа с VCS

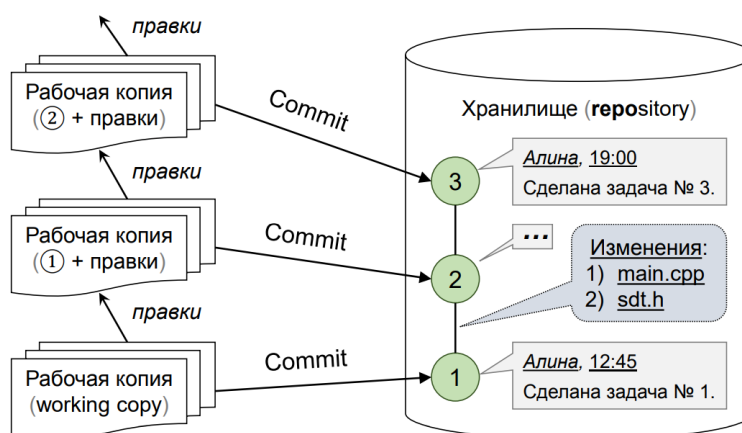


Рис. 5.1: .

5. Опишите порядок работы с общим хранилищем VCS. (рис. [5.2])

## Работа с общим хранилищем

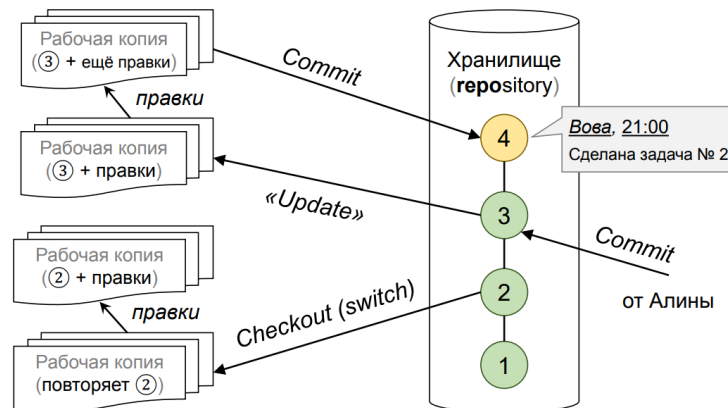


Рис. 5.2: .

6. Каковы основные задачи, решаемые инструментальным средством git? У Git две основных задачи: первая — хранить информацию о всех изменениях в вашем коде, начиная с самой первой строчки, а вторая — обеспечение удобства командной работы над кодом.
7. Назовите и дайте краткую характеристику командам git. git init - создание репозитория git add (имена файлов) - Добавляет файлы в индекс git commit – выполняет коммит проиндексированных файлов в репозиторий git status – показывает какие файлы изменились между текущей стадией и HEAD. Файлы разделяются на 3 категории: новые файлы, измененные файлы, добавленные новые файлы git checkout (sha1 или метка) - получение указанной версии файла git push – отправка изменений в удаленный репозиторий git fetch – получение изменений из удаленного репозитория git clone (remote url) - клонирование удаленного репозитория себе
8. Приведите примеры использования при работе с локальным и удалённым репозиториями. (рис. [5.3])

```
[knikulina@fedora study_2022-2023_os-intro]$ git add .  
[knikulina@fedora study_2022-2023_os-intro]$ git commit -am 'feat(main): make course structure'
```

Рис. 5.3: .

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

Ветка (англ. branch) — это последовательность коммитов, в которой ведётся параллельная разработка какого-либо функционала. Основная ветка — master. Ветки в GIT. Показать все ветки, существующие в репозитории `git branch`. Создать ветку `git branch имя`.

Ветки нужны, чтобы несколько программистов могли вести работу над одним и тем же проектом или даже файлом одновременно, при этом не мешая друг другу. Кроме того, ветки используются для тестирования экспериментальных функций: чтобы не повредить основному проекту, создается новая ветка специально для экспериментов.

## 10. Как и зачем можно игнорировать некоторые файлы при commit? Игнорируемые файлы — это, как правило, артефакты сборки и файлы, генерируемые машиной из исходных файлов в вашем репозитории, либо файлы, которые по какой-либо иной причине не должны попадать в коммиты. Вот некоторые распространенные примеры таких файлов:

кэши зависимостей, например содержимое `node_modules` или `packages`; скомпилированный код, например файлы `.o`, `.рус` и `.class`; каталоги для выходных данных сборки, например `bin`, `out` или `target`; файлы, сгенерированные во время выполнения, например `.log`, `.lock` или `.tmp`; скрытые системные файлы, например `.DS_Store` или `Thumbs.db`; личные файлы конфигурации IDE, например `.idea.workspace.xml`.