

# **Отчет по лабораторной работе №2**

**Первоначальная настройка git**

Галацан Николай, НПИбд-01-22

# Содержание

1	Цель работы	4
2	Теоретическое введение	5
3	Выполнение лабораторной работы	6
4	Выводы	13
5	Ответы на контрольные вопросы	14

## Список иллюстраций

3.1	Аккаунт на Github . . . . .	6
3.2	Предварительная конфигурация git . . . . .	7
3.3	Генерация ключей . . . . .	7
3.4	Созданный SSH ключ на сайте . . . . .	8
3.5	Создание ключа GPG . . . . .	8
3.6	Копирование ключа GPG . . . . .	9
3.7	Созданный ключ GPG на сайте . . . . .	9
3.8	Настройка подписей коммитов git . . . . .	9
3.9	Настройка gh . . . . .	10
3.10	Создание каталога для предмета и клонирование репозитория . .	11
3.11	Настройка каталога курса . . . . .	11
3.12	Проверка репозитория на странице github . . . . .	12

# 1 Цель работы

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

## 2 Теоретическое введение

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом.

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.

Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями.

Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

### 3 Выполнение лабораторной работы

Устанавливаю git: `dnf install git`

Создаю учетную запись на сайте репозитория Github (<https://github.com/>) и заполняю основные данные (рис. 3.1)

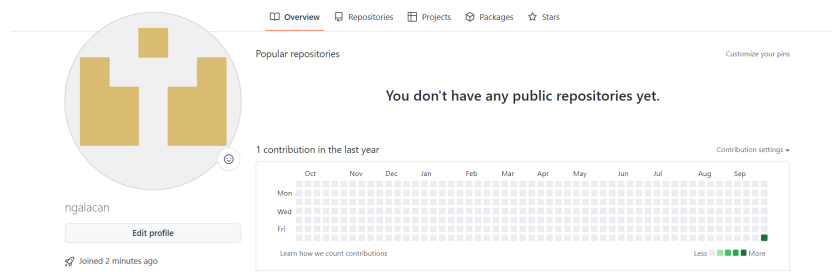


Рис. 3.1: Аккаунт на Github

Делаю предварительную конфигурацию git (рис. 3.2). Открываю терминал и ввожу команды, указав имя и email владельца репозитория:

```
git config --global user.name "ngalacan"
```

```
git config --global user.email "1032225763@pfur.ru"
```

Настраиваю utf-8 в выводе сообщений git:

```
git config --global core.quotePath false
```

Задаю имя начальной ветки (master):

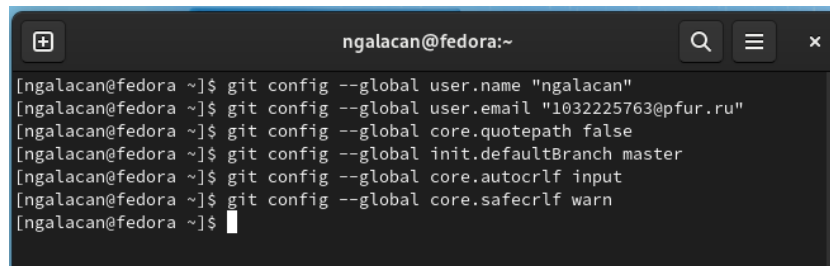
```
git config --global init.defaultBranch master
```

Параметр autocrlf:

```
git config --global core.autocrlf input
```

Параметр safecrlf:

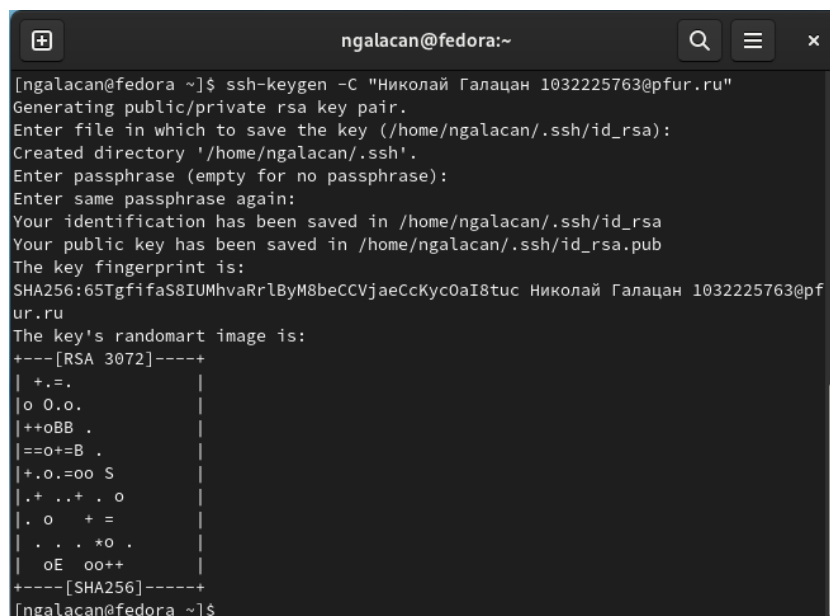
```
git config --global core.safecrlf warn
```

A terminal window titled 'ngalacan@fedora:~' showing a series of git configuration commands being executed. The commands set the global user name to 'ngalacan', the email to '1032225763@pfur.ru', disable quote path, set the default branch to 'master', set autocrlf to 'input', and set safecrlf to 'warn'.

```
ngalacan@fedora:~  
[ngalacan@fedora ~]$ git config --global user.name "ngalacan"  
[ngalacan@fedora ~]$ git config --global user.email "1032225763@pfur.ru"  
[ngalacan@fedora ~]$ git config --global core.quotePath false  
[ngalacan@fedora ~]$ git config --global init.defaultBranch master  
[ngalacan@fedora ~]$ git config --global core.autocrlf input  
[ngalacan@fedora ~]$ git config --global core.safecrlf warn  
[ngalacan@fedora ~]$
```

Рис. 3.2: Предварительная конфигурация git

Генерирую ключи SSH для идентификации пользователя на сервере репозитория (рис. 3.3).

A terminal window titled 'ngalacan@fedora:~' showing the process of generating an SSH key pair using the 'ssh-keygen' command. It prompts for a file name, creates a directory, asks for a passphrase, and displays the key fingerprint and a randomart image.

```
ngalacan@fedora:~  
[ngalacan@fedora ~]$ ssh-keygen -C "Николай Галацан 1032225763@pfur.ru"  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/ngalacan/.ssh/id_rsa):  
Created directory '/home/ngalacan/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/ngalacan/.ssh/id_rsa  
Your public key has been saved in /home/ngalacan/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:65TgfifaS8IUMhvaRrlByM8beCCVjaeCcKyc0aI8tuc Николай Галацан 1032225763@pfur.ru  
The key's randomart image is:  
+---[RSA 3072]-----+  
| +. =. |  
| o 0.o. |  
| ++oBB . |  
| ==o+=B . |  
| +.o.=oo S |  
| ,+ ..+ . o |  
| . o + = |  
| . . . *O . |  
| oE oo++ |  
+---[SHA256]-----+  
[ngalacan@fedora ~]$
```

Рис. 3.3: Генерация ключей

Загружаю открытый ключ на Github. На сайте перехожу *Setting => SSH and GPG keys => New SSH key*. Ввожу в терминал `cat ~/.ssh/id_rsa.pub | xclip -sel clip` и копирую ключ. Вставляю в появившееся на сайте поле скопированный ключ и указываю имя (рис. 3.4).

## SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

### Authentication Keys



#### New SSH

SHA256:65TgfifaS8IUHvaRr1BvY8BeCCVjaeCcKyc0aI8tuc

Added on 5 Oct 2022

Never used — Read/write

[Delete](#)

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

Рис. 3.4: Созданный SSH ключ на сайте

Создаю ключи GPG с помощью команды `gpg --full-generate-key` и выбираю из предложенных опций тип *RSA and RSA*, размер *4096*; срок действия - значение по умолчанию — 0. Заполняю личную информацию (рис. 3.5).

```
ngalacan@fedora:~ — gpg --full-generate-key
[ngalacan@ngalacan ~]$ gpg --full-generate-key
gpg (GnuPG) 2.3.7; 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.

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
  <n>y = срок действия ключа - n лет
Срок действия ключа? (0)
```

Рис. 3.5: Создание ключа GPG

Добавляю ключ GPG на github, предварительно выведя список ключей и скопировав отпечаток приватного ключа (рис. 3.6, 3.7).



```
[ngalacan@ngalacan ~]$ 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/ngalacan/.gnupg/pubring.kbx
-----
sec   rsa4096/65FA3CE4164D9FDA 2023-02-12 [SC]
      C5142C2712175F832435AC2F65FA3CE4164D9FDA
uid    [ абсолютно ] ngalacan <1032225763@pfur.ru>
ssb    rsa4096/F41AB8A129B93212 2023-02-12 [E]

[ngalacan@ngalacan ~]$ gpg --armor --export <PGP Fingerprint> | xclip -sel clip
bash: синтаксическая ошибка рядом с неожиданным маркером «|»
[ngalacan@ngalacan ~]$ gpg --armor --export <PGP Fingerprint> | xclip -sel clip
bash: синтаксическая ошибка рядом с неожиданным маркером «|»
[ngalacan@ngalacan ~]$ gpg --armor --export 65FA3CE4164D9FDA | xclip -sel clip
[ngalacan@ngalacan ~]$
```

Рис. 3.6: Копирование ключа GPG

### GPG keys / Add new

Title

New GPG

Key

```
tpLO3/f67GILp8MBpg0Qo+mvf9x4GoMKQLB5xhCKEUovGGIke7vveLCLMu04A6Vo
zWQo6lm4QG16OFy+z87d7deSviMsvQZPGE6l0j5C2vLMui49gIK0SDuuxnSSNqz
Eq7M/qeJjueFEs1cnTJZZWQ8AK4uvBjJrG+YHYr7BXc0Bq5cmeu0mARaEqIU3ZP
/VIQ5SDvCIgaGdW1x2CvKxLFi47V07gudiTzsjycm9k5uiF5bmR0Me2elmi+KVy
xfuPkt67Nq5jfwVqeQfTQ3h0oCZgqLCZWrpvmH/1a2CWYhvKVslWtLWbt7fODOca
yHOJEURAOik1+qutGWtIS92nPa4gCjsCrwCHZd4uo8VZU90P0W/Xxr7yJg=
=HDJQ
-----END PGP PUBLIC KEY BLOCK-----
```

Add GPG key

Рис. 3.7: Созданный ключ GPG на сайте

Настраиваю автоматические подписи коммитов git (рис. 3.8).

```
[ngalacan@ngalacan ~]$ git config --global user.signingkey 65FA3CE4164D9FDA
[ngalacan@ngalacan ~]$ git config --global commit.gpgsign true
[ngalacan@ngalacan ~]$ git config --global gpg.program $(which gpg2)
[ngalacan@ngalacan ~]$
```

Рис. 3.8: Настройка подписей коммитов git

Устанавливаю и настраиваю gh для авторизации в github, привязываю устройство (рис. 3.9).

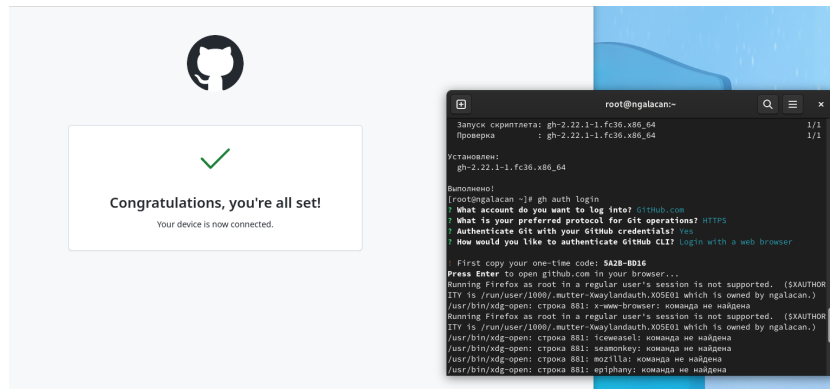
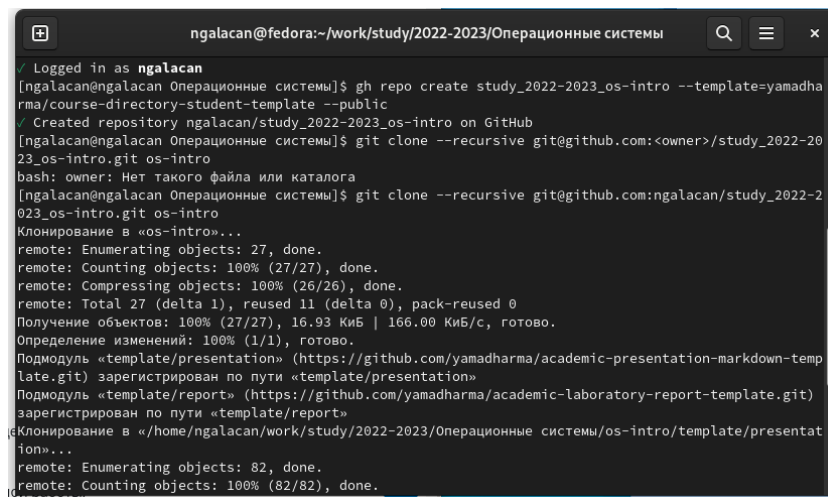


Рис. 3.9: Настройка gh

Создаю через терминал каталог для предмета “Операционные системы” и клонирую репозиторий с помощью команд

```
mkdir -p ~/work/study/2022-2023/"Операционные системы"
cd ~/work/study/2022-2023/"Операционные системы"
gh repo create study_2022-2023_os-intro --template=yamadharma/course-
directory-student-template --public
git clone --recursive git@github.com:<owner>/study_2022-2023_os-
intro.git os-intro
```

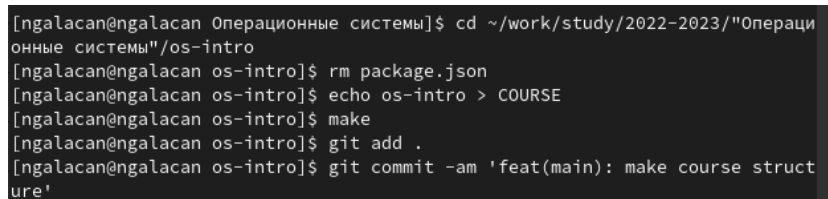
(рис. 3.10).



```
ngalacan@fedora:~/work/study/2022-2023/Операционные системы
✓ Logged in as ngalacan
[ngalacan@ngalacan Операционные системы]$ gh repo create study_2022-2023_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository ngalacan/study_2022-2023_os-intro on GitHub
[ngalacan@ngalacan Операционные системы]$ git clone --recursive git@github.com:<owner>/study_2022-2023_os-intro.git os-intro
bash: owner: Нет такого файла или каталога
[ngalacan@ngalacan Операционные системы]$ git clone --recursive git@github.com:ngalacan/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 | 166.00 КиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/ngalacan/work/study/2022-2023/Операционные системы/os-intro/template/presentation»...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
```

Рис. 3.10: Создание каталога для предмета и клонирование репозитория

Перехожу в каталог курса и удаляю лишние файлы, используя команды `cd` и `rm`. Создаю необходимые каталоги и отправляю файлы на сервер (рис. 3.11).



```
[ngalacan@ngalacan Операционные системы]$ cd ~/work/study/2022-2023/"Операционные системы"/os-intro
[ngalacan@ngalacan os-intro]$ rm package.json
[ngalacan@ngalacan os-intro]$ echo os-intro > COURSE
[ngalacan@ngalacan os-intro]$ make
[ngalacan@ngalacan os-intro]$ git add .
[ngalacan@ngalacan os-intro]$ git commit -am 'feat(main): make course structure'
```

Рис. 3.11: Настройка каталога курса

Проверяю, отправились ли файлы на github (рис. 3.12).


















 <b>ngalacan</b> feat(main): make course structure <span>4a97511 2 minutes ago</span> <span> 2 commits</span>		
 config	Initial commit	6 minutes ago
 labs	feat(main): make course structure	2 minutes ago
 presentation	feat(main): make course structure	2 minutes ago
 project-personal	feat(main): make course structure	2 minutes ago
 template	Initial commit	6 minutes ago
 .gitattributes	Initial commit	6 minutes ago
 .gitignore	Initial commit	6 minutes ago
 .gitmodules	Initial commit	6 minutes ago
 CHANGELOG.md	Initial commit	6 minutes ago
 COURSE	feat(main): make course structure	2 minutes ago
 LICENSE	Initial commit	6 minutes ago
 Makefile	Initial commit	6 minutes ago
 README.en.md	Initial commit	6 minutes ago
 README.git-flow.md	Initial commit	6 minutes ago
 README.md	Initial commit	6 minutes ago

Рис. 3.12: Проверка репозитория на странице github

Файлы соответствуют.

## 4 Выводы

Была изучена идеология и применение средств контроля версий. Была настроена система git. Приобретены практические навыки по работе с системой git.

## 5 Ответы на контрольные вопросы

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

Системы контроля версий (Version Control System, VCS) - система регистрации изменений в одном или нескольких файлах. Применяются при работе нескольких человек над одним проектом.

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

Хранилище - область на локальном или удаленном репозитории, где хранятся файлы проекта.

Commit - внесение правок или изменение файлов в репозитории.

История - история изменений в репозитории.

Рабочая копия - копия главного удаленного репозитория, в которой работает пользователь.

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

Централизованная модель предполагает наличие единого репозитория для хранения файлов. В отличие от централизованных, в распределённых (децентрализованных) системах контроля версий центральный репозиторий не является обязательным. Примеры централизованных VCS: CVS, Subversion; примеры децентрализованных: Git, Bazaar.

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

Сохранение файлов на удаленном репозитории, коммит, отправка на сервер и др.

5. Опишите порядок работы с общим хранилищем VCS.

Обновление до текущей версии репозитория (например, `git pull`), сохранение файлов на удаленном репозитории, коммит, отправка на сервер и др.

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

Объединение изменений, выбор нужной версии, отмена изменений или блокировка и др.

7. Назовите и дайте краткую характеристику командам `git`.

Создание основного дерева репозитория: `git init`

Получение обновлений 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`

добавить все изменённые и/или созданные файлы и/или каталоги: `git add .`

сохранить все добавленные изменения и все изменённые файлы: `git commit -am 'Описание коммита'`

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

`git pull` - для обновления локального репозитория.

`git status` - для просмотра изменений локального репозитория.

`git add .` - для добавления всех изменений в удаленный репозиторий.

`git push` - для отправки всех изменений на удаленный репозиторий.

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

Ветви - последовательность коммитов и изменений. Они нужны для работы нескольких пользователей над одним проектом.

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

С помощью команды `git rm имена_файлов`. Это нужно для того, чтобы файл не отправился в удаленный репозиторий (например, если работа не завершена или он не нужен, то и в коммит он попасть не должен).