

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

## Факультет физико-математических и естественных наук

### Отчёт по лабораторной работе №2

#### Управление версиями

**Дисциплина:** Операционные системы

**Студент:** Оразгелдиева Огулнур

**Группа:** НПИбд-02-20

---

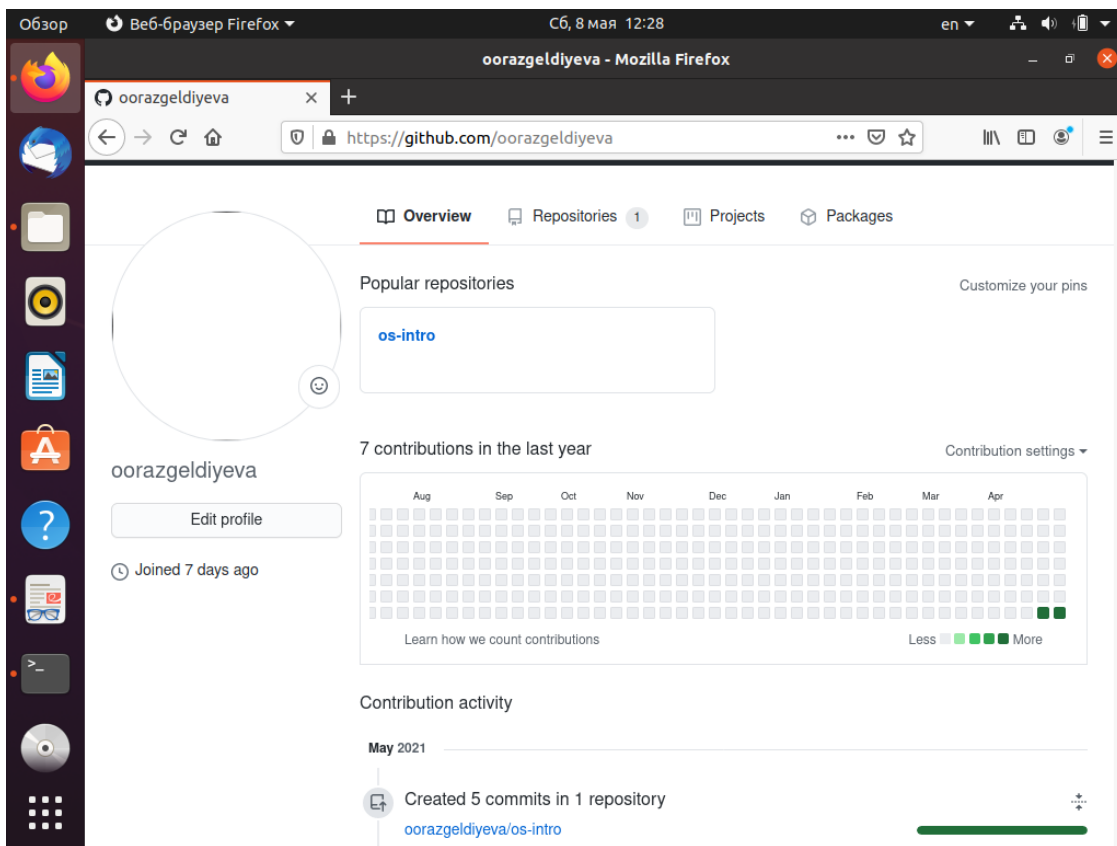
#### Лабораторная работа №2

##### Управление версиями

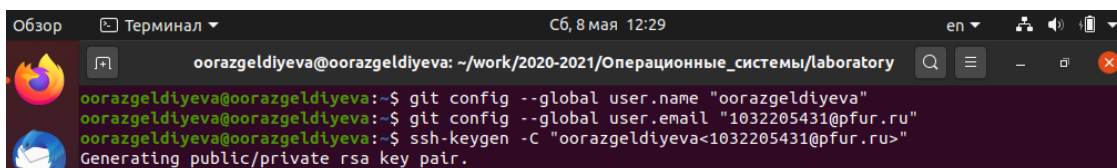
**Цель работы:** изучить идеологию и применение средств контроля версий

**Ход работы:**

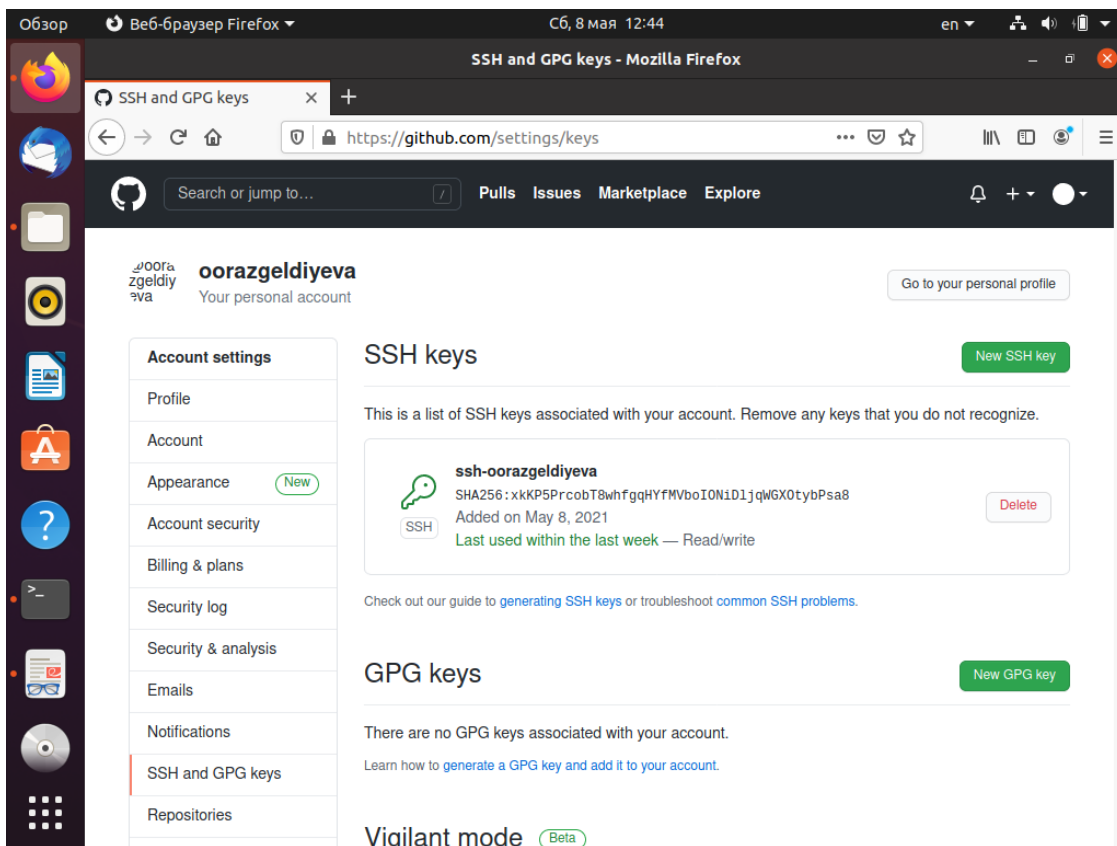
1. Создала учетную запись на github. [github.com] (<https://github.com/oorazgeldiyeva>)



2. Сделала конфигурацию с помощью команды `git config`, потом создала ssh-ключ, и скопировала его в буфер обмена, вставила на github



```
Обзор Терминал C6, 8 мая 12:29 en
oorazgeldiyeva@oorazgeldiyeva: ~/work/2020-2021/Операционные_системы/laboratory
oorazgeldiyeva@oorazgeldiyeva:~$ git config --global user.name "oorazgeldiyeva"
oorazgeldiyeva@oorazgeldiyeva:~$ git config --global user.email "1032205431@pfur.ru"
oorazgeldiyeva@oorazgeldiyeva:~$ ssh-keygen -C "oorazgeldiyeva<1032205431@pfur.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/oorazgeldiyeva/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match. Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in
Your public key has been saved in .pub
The key fingerprint is:
SHA256:s9LZxXqb6CQTuI8FuRY3KEn01jWbbih+/RE1QtqmNDE oorazgeldiyeva<1032205431@pfur.ru>
The key's randomart image is:
+----[RSA 3072]-----+
  o . E .
+ o X .
o . *o+ o o
o +=+oo +
. oS=o.o .
. oo*= o o
. . =.* o o
. . o *
. . oo .
+-----[SHA256]-----+
oorazgeldiyeva@oorazgeldiyeva:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDQZS0Uaxhw5f51VP6ndgT42kC4DwACw3bTFRXVSf8TpiL+PeTdLZRmJ501fGd0q+9b
cwfr+BPFCro+L7BZ158KM7gKaTF84dB0i6h0GCMhmpdluToVhnj1d+cwYMsAIUkIq7AoEe/31DE+6Ajgm7USAs9bKrZ2vxNMvONE5Y/
mVC4XMGxryUyhSVypSCK651sE4e2soRdtI7emdXhKZyqw380tgWKnJMRGgEBcem0sEopm1EckKxJz4xJ2m4zFH85K+QDwEcI8XBUIroM
bgnM0jKKHVkjDy002LuLRb0WS5VEXC2UQUr0f3CS4Dwo+oL01q4rThrfD2k9rn43BDcJGuFR3B9gsxwA0tlyIAVeTrP/NqFuUDV/+2rm
85nyk3pxBNKQ0iL7B/fIFlnBb/AGDrUh7gfzFxeLLCYb508F8ILzkntXAqkpihMsqu8sAMH5W1dMhL6MDeBNnLQZQZ2u218dwyMRot
j6AycAb0aLAgJAvXJB4ZL60wDMG3XWs= oorazgeldiyeva<1032205431@pfur.ru>
oorazgeldiyeva@oorazgeldiyeva:~$ mkdir work
oorazgeldiyeva@oorazgeldiyeva:~$ cd work
oorazgeldiyeva@oorazgeldiyeva:~/work$ mkdir 2020-2021
oorazgeldiyeva@oorazgeldiyeva:~/work$ cd 2020-2021
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021$ mkdir Операционные_системы
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021$ cd Операционные_системы
```



3. Создала каталог файлов согласно пункту M2

```
Обзор Терминал C6, 8 мая 12:46 en
oorazgeldiyeva@oorazgeldiyeva: ~/work/2020-2021/Операционные_системы/laboratory

Passphrases do not match. Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in
Your public key has been saved in .pub
The key fingerprint is:
SHA256:s9LzXqb6CQ8Fui8FuRY3KEno1jWbbih+/RE1QtqmNDE oorazgeldiyeva<1032205431@pfur.ru>
The key's randomart image is:
+----[RSA 3072]-----+
|
| o .E .
| + o X .
| o . *o+ o o
| o+=+oo +
| .oS=O.o .
| . oo*.= o o
| . .=. * o o
| . .o *
| . oo .
+----[SHA256]-----+
oorazgeldiyeva@oorazgeldiyeva:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDzZSOUaxhW5fs1VP6ndgT42kC4DwACw3bTFRXVSf8TpIL+PeTdLZRmJ501fGd0q+9b
cwF+BPFCro+l7BZ158KM7gKaTF84dBRoI6h0GCMhmpdLuToVhnj1d+cwYMsAIUkIq7AoEe/31DE+6A9m7USAs9bKrZ2vxNMvONE5Y/
mVC4XMGxryUyhSVypSCK651sE4e2soRdti7emdXhKZyqw380tgWKnJMRGgEBcem0sEopm1EckKxJz4xJ2m4zFH85K+QDwEcI8XBUIroM
bgnM0jXKHVKjdy002LuLRb0WS5VEXC2UQUr0f3CS4Dwo+oL01q4rTHrfd2k9rn43BDcJGuFR3B9gsxwA0tiyIAVeTrP/NqFuUDV/+2rm
85nyk3pxBNKQ0iiG7B/fIFln8Bb/AGDrUh7gfzFxeLLlCYb508F8IlzkmtXAqpihMsqu8sAMH5W1dMhL6Mde8NniQZQz2u218dwyMROT
j6AycAb0aLagJAvXJB4Zl60wDMG3XWs= oorazgeldiyeva<1032205431@pfur.ru>
oorazgeldiyeva@oorazgeldiyeva:~$ mkdir work
oorazgeldiyeva@oorazgeldiyeva:~$ cd work
oorazgeldiyeva@oorazgeldiyeva:~/work$ mkdir 2020-2021
oorazgeldiyeva@oorazgeldiyeva:~/work$ cd 2020-2021
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021$ mkdir Операционные_системы
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021$ cd Операционные_системы
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы$ mkdir laboratory
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы$ cd laboratory
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git init
Инициализирован пустой репозиторий Git в /home/oorazgeldiyeva/work/2020-2021/Операционные_системы/labora
tory/.git/
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ echo "# Лабораторные раб
оты">> README.md
```

4. Инициализировала git, сделав первый коммит, выложила файл README.md на github. Использовала команды echo (для создания «заготовки» для коммита), git add (добавление), git commit (для коммита), git push (выложить на гит)

```
Обзор Терминал C6, 8 мая 12:54 en
oorazgeldiyeva@oorazgeldiyeva: ~/work/2020-2021/Операционные_системы/laboratory
oorazgeldiyeva@oorazgeldiyeva:~/work$ cd 2020-2021
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021$ mkdir Операционные_системы
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021$ cd Операционные_системы
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы$ mkdir laboratory
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы$ cd laboratory
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git init
Инициализирован пустой репозиторий Git в /home/oorazgeldiyeva/work/2020-2021/Операционные_системы/laboratory/.git/
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ echo "# Лабораторные работы">> README.md
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git add README.md
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git commit -a "first commit"
fatal: paths 'first commit ...' with -a does not make sense
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git commit -m "first commit"
[master (корневой коммит) 53b45c9] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git remote add origin
использование: git remote add [<опции>] <имя> <url>

-f, --fetch           извлечь внешние ветки
--tags               импортировать все метки и ассоциированные объекты при извлечении
                    или не извлекать метки вообще (--no-tags)
-t, --track <ветка>  отслеживаемые ветки
-m, --master <ветка> мастер ветка
--mirror[=(push|fetch)]
                    настроить внешний репозиторий как зеркало для отправки или извлечения изменений

oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ ↵
↵ команда не найдена
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git remote add origin git@github.com:oorazgeldiyeva/os-intro.git
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git push -u origin master
Warning: Permanently added the RSA host key for IP address '140.82.121.3' to the list of known hosts.
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 247 bytes | 11.00 KiB/s, готово.
Всего 3 (изменения 0), повторно использовано 0 (изменения 0)
To github.com:oorazgeldiyeva/os-intro.git
* [new branch]      master -> master
Ветка «master» отслеживает внешнюю ветку «master» из «origin».
```

```
Обзор Терминал C6, 8 мая 12:54 en
oorazgeldiyeva@oorazgeldiyeva: ~/work/2020-2021/Операционные_системы/laboratory
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ echo "# Лабораторные работы">> README.md
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git add README.md
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git commit -a "first commit"
fatal: paths 'first commit ...' with -a does not make sense
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git commit -m "first commit"
[master (корневой коммит) 53b45c9] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git remote add origin
использование: git remote add [<опции>] <имя> <url>

-f, --fetch           извлечь внешние ветки
--tags               импортировать все метки и ассоциированные объекты при извлечении
                    или не извлекать метки вообще (--no-tags)
-t, --track <ветка>  отслеживаемые ветки
-m, --master <ветка> мастер ветка
--mirror[=(push|fetch)]
                    настроить внешний репозиторий как зеркало для отправки или извлечения изменений

oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ ↵
↵ команда не найдена
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git remote add origin git@github.com:oorazgeldiyeva/os-intro.git
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git push -u origin master
Warning: Permanently added the RSA host key for IP address '140.82.121.3' to the list of known hosts.
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 247 bytes | 11.00 KiB/s, готово.
Всего 3 (изменения 0), повторно использовано 0 (изменения 0)
To github.com:oorazgeldiyeva/os-intro.git
* [new branch]      master -> master
Ветка «master» отслеживает внешнюю ветку «master» из «origin».
```

5. Скачала файл лицензии, загрузила шаблон игнорируемых файлов для C и выложила все на github

```
Обзор Терминал C6, 8 мая 12:57 en
oorazgeldiyeva@oorazgeldiyeva: ~/work/2020-2021/Операционные_системы/laboratory

Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 247 bytes | 11.00 KiB/s, готово.
Всего 3 (изменения 0), повторно использовано 0 (изменения 0)
To github.com:oorazgeldiyeva/os-intro.git
* [new branch] master -> master
Ветка «master» отслеживает внешнюю ветку «master» из «origin».
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O LICENSE
--2021-05-08 11:59:34-- https://creativecommons.org/licenses/by/4.0/legalcode.txt
Распознаётся creativecommons.org (creativecommons.org)... 172.67.34.140, 104.20.151.16, 104.20.150.16, ...
Подключение к creativecommons.org (creativecommons.org)[172.67.34.140]:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: нет данных [text/plain]
Сохранение в: «LICENSE»

LICENSE [ <=> ] 18,22K ---KB/s за 0,04s

2021-05-08 11:59:36 (406 KB/s) - «LICENSE» сохранён [18657]

oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ curl -L -s https://www.gitignore.io/api/list
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ curl -L -s https://www.gitignore.io/api/c >> .gitignore
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git add .
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git commit -am "commit"
[master 3a87202] commit
2 files changed, 396 insertions(+)
create mode 100644 .gitignore
create mode 100644 LICENSE
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (4/4), 6.08 KiB | 222.00 KiB/s, готово.
Всего 4 (изменения 0), повторно использовано 0 (изменения 0)
To github.com:oorazgeldiyeva/os-intro.git
53b45c9..3a87202 master -> master
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git flow init
```

6. Сначала установила git-flow с помощью `sudo apt-get install git-flow`. Инициализировала gitflow (префикс ярлыков установила в `v`), убедилась, что нахожусь на ветке `develop`

```
Обзор Терминал C6, 8 мая 13:02 en
oorazgeldiyeva@oorazgeldiyeva: ~/work/2020-2021/Операционные_системы/laboratory

To github.com:oorazgeldiyeva/os-intro.git
53b45c9..3a87202 master -> master
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git flow init
Which branch should be used for bringing forth production releases?
- master
Branch name for production releases: [master]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? [v]
Hooks and filters directory? [/home/oorazgeldiyeva/work/2020-2021/Операционные_системы/laboratory/.git/hooks]
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git branch
* develop
master
```

7. Создала релиз с версией 1.0.0



```
Обзор Терминал C6, 8 мая 13:09 en
oorazgeldiyeva@oorazgeldiyeva: ~/work/2020-2021/Операционные_системы/laboratory

master
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git flow release start 1.0.0
Переключено на новую ветку «release/1.0.0»

Summary of actions:
- A new branch 'release/1.0.0' was created, based on 'develop'
- You are now on branch 'release/1.0.0'

Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

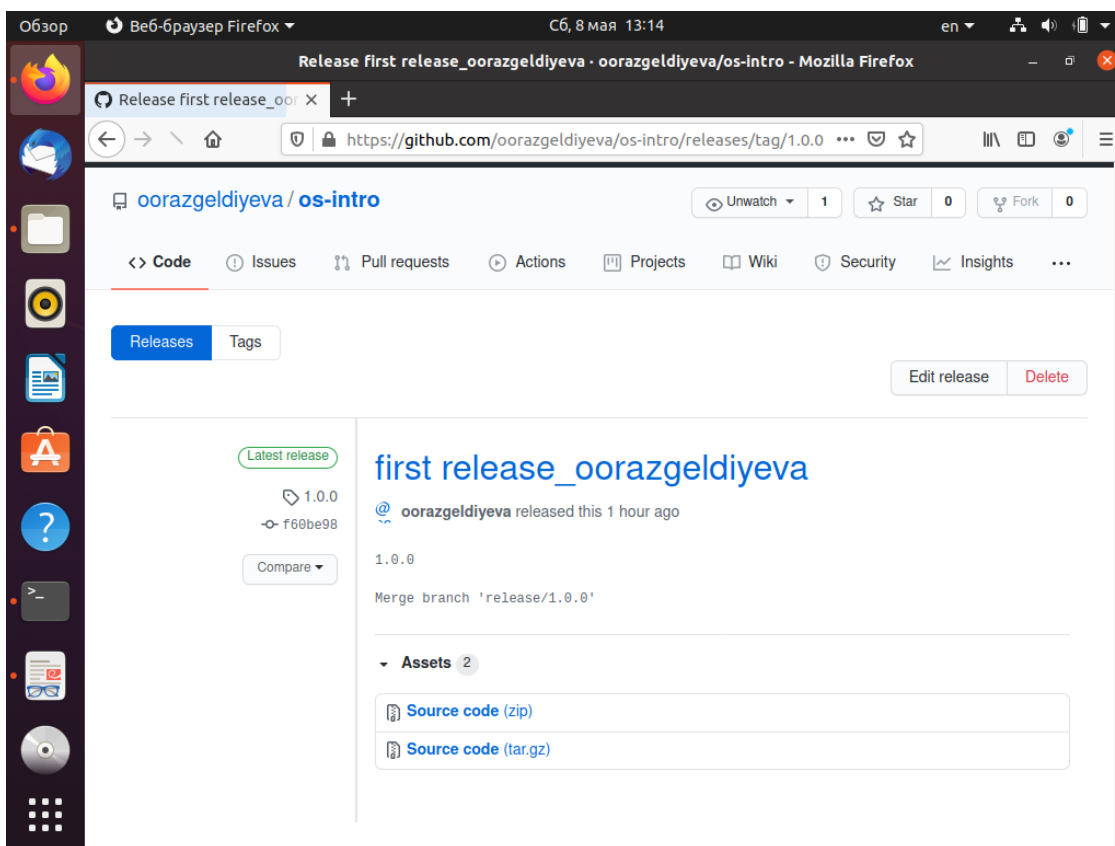
    git flow release finish '1.0.0'

oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ echo "1.0.0" >> VERSION
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git add .
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git commit -am 'chore(main): add version'
[release/1.0.0 7be8ca1] chore(main): add version
1 file changed, 1 insertion(+)
create mode 100644 VERSION
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git flow release finish 1.0.0
Переключено на ветку «master»
Ваша ветка обновлена в соответствии с «origin/master».
Merge made by the 'recursive' strategy.
VERSION | 1 +
1 file changed, 1 insertion(+)
create mode 100644 VERSION
Уже на «master»
Ваша ветка опережает «origin/master» на 2 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
fatal: нет описания метки?
fatal: Tagging failed. Please run finish again to retry.
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git push --all
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
```

```
Обзор Терминал C6, 8 мая 13:11 en
oorazgeldiyeva@oorazgeldiyeva: ~/work/2020-2021/Операционные_системы/laboratory

fatal: Tagging failed. Please run finish again to retry.
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git push --all
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (4/4), 379 bytes | 189.00 KiB/s, готово.
Всего 4 (изменения 2), повторно использовано 0 (изменения 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:oorazgeldiyeva/os-intro.git
  3a87202..f60be98  master -> master
  * [new branch]    develop -> develop
  * [new branch]    release/1.0.0 -> release/1.0.0
oorazgeldiyeva@oorazgeldiyeva:~/work/2020-2021/Операционные_системы/laboratory$ git push --tags
error: unknown option 'tags'
использование: git push [<опции>] [<репозиторий> [<спецификатор-ссылки>...]]

-v, --verbose      быть многословнее
-q, --quiet        тихий режим
--repo <репозиторий> репозиторий
--all              отправить все ссылки
--mirror           сделать зеркало всех ссылок
-d, --delete       удалить ссылки
--tags            отправить метки (нельзя использовать вместе с --all или --mirror)
-n, --dry-run      пробный запуск
--porcelain        машиночитаемый вывод
-f, --force        принудительное обновление
--force-with-lease[=<refname>:<экспресс>]
                  требовать, чтобы старое значение ссылки было ожидаемым
--recurse-submodules[=(check|on-demand|no)]
                  управление рекурсивной отправкой подмодулей
--thin             использовать тонкие пакеты
--receive-pack <receive-pack>
                  путь к программе упаковки на сервере
--exec <receive-pack>
                  путь к программе упаковки на сервере
-u, --set-upstream установить вышестоящую ветку для git pull/status
--progress         принудительно выводить прогресс
--prune            почистить локально удаленные ссылки
--no-verify        пропустить перехватчик pre-push
--follow-tags       следовать за тегами, если они появились
```



## Вывод:

на лабораторной работе изучила идеологию и применение средств контроля версий.

## Контрольные вопросы:

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

### Ответ:

Системы контроля версий (VCS) - программное обеспечение для облегчения работы с изменяющейся информацией, позволяющее хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям.

Предназначены для работы нескольких человек над одним проектом, а также при разработке программного обеспечения для хранения исходных кодов разрабатываемой программы.

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

**Ответ:** Хранилище – место «памяти», в котором будет храниться новая версия файла после его изменения пользователем.



Commit. В нем содержится описание тех изменений, которые вносит пользователь в код приложения.

История – история изменений. Обычно доступна информация о том, кто из участников, когда и какие изменения вносил.

Рабочая копия – это копия, которую мы выписали в свою рабочую зону, это то, над чем мы работаем в данный момент. Привилегированный доступ только одному пользователю, работающему с файлом

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

**Ответ:**

Централизованные VCS предполагают наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Пример: AccuRev Децентрализованные VCS не имеют единого репозитория, он у каждого пользователя свой. Помимо того, они были созданы для обмена изменениями, а не для их объединения. Не имеют какой-то жестко заданной структуры репозитория с центральным сервером. Пример: Git

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

**Ответ:**

При единоличной работе с VCS каждое новое изменение в репозитории сохраняется не со всеми предыдущими версиями. Оно изменяется по системе: одно предыдущее + новая информация

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

**Ответ:**

1. Создать репозиторий. Это место, где будут лежать файлы. Теперь у нас есть общее хранилище данных, с которым и будет проходить дальнейшая работа.
2. Скачать проект из репозитория
3. Обновить проект, забрать последнюю версию из репозитория
4. Внести изменения в проект
5. Запустить код, т.е изменить код в общем хранилище
6. Создать ветку

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

**Ответ:**

1. Сохранение файлов с исходным кодом
2. Защита от случайных исправлений и удалений
3. Отмена изменений и удалений, если они оказались некорректными
4. Одновременная поддержка рабочей версии и разработка новой
5. Возврат к любой версии кода из прошлого
6. Просмотр истории изменений

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

**Ответ:**

*git init* – создание основного дерева репозитория

*git push* – отправка всех произведённых изменений локального дерева в центральный репозиторий

*git status* – просмотр списка изменённых файлов в текущей директории

*git add* – добавить все изменённые и/или созданные файлы и/или каталоги

*git add имена\_файлов* – добавить конкретные изменённые и/или созданные файлы и/или каталоги

*git pull* – получение обновлений (изменений) текущего дерева из центрального репозитория

*git rm имена\_файлов* – удалить файл и/или каталог из индекса репозитория

*git commit -am 'Описание коммита'* – сохранить все добавленные изменения и все изменённые файлы

*git commit* – сохранить добавленные изменения с внесением комментария через встроенный редактор

*git checkout -b имя\_ветки* – создание новой ветки, базирующейся на текущей: *git checkout имя\_ветки* – переключение на ветку

*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 удаление локальной ветки

*git push origin :имя\_ветки* – удаление ветки с центрального репозитория

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

**Ответ:**

Локальный репозиторий – она же директория “.git”. В ней хранятся коммиты и другие объекты. Локальный репозиторий мы используем, когда работаем одни и нам нужно сохранить свои же изменения.

Удаленный репозиторий – тот репозиторий, который считается общим, в который мы можем передать свои коммиты из локального репозитория, чтобы остальные пользователи могли их увидеть. Удаленный репозиторий используется для групповой работы, когда в личном репозитории скопилось достаточно коммитов, мы делимся ими

в удаленном для того, чтобы другие пользователи могли видеть наши изменения. Также из удаленного репозитория мы можем скачать чужие изменения.

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

### **Ответ**

Ветка – это подвижный указатель на один из коммитов. Обычно ветка указывает на последний коммит в цепочке коммитов. В ветке мы можем как угодно ломать проект, основной код при этом не пострадает.

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

### **Ответ:**

Игнорируемые файлы – это, как правило, специфичные для платформы файлы или автоматически созданные файлы из систем сборки.

.gitignore — это простой текстовый файл, в каждой строке которого содержится шаблон, который файлы или каталоги следует игнорировать. Он использует шаблоны подстановки для сопоставления имен файлов с подстановочными знаками