

Шаблон отчёта по лабораторной работе

Судаков Егор Андореевич

List of Tables

List of Figures

1 Цель работы

Изучить идеологию и применение средств контроля версий, изучить работу с репозиториями и сайтом Github

2 Задание

2.0.1 1. Настройка git

1. Создайте учётную запись на <https://github.com>.
2. Настройте систему контроля версий git, как это описано выше с использованием сервера репозитория
3. Создайте структуру каталога лабораторных работ согласно пункту М.2 ###
2. Подключение репозитория к github
 - Создайте репозиторий на GitHub. Для примера назовём egoos-intro.
 - Рабочий каталог будем обозначать как laboratory. Вначале нужно перейти в этот каталог: `cd laboratory`
 - Инициализируем системы `git:git init`
 - Создаём заготовку для файла README.md: `echo "# Лабораторные работы" > README.md`
 - Делаем первый коммит и выкладываем на github:

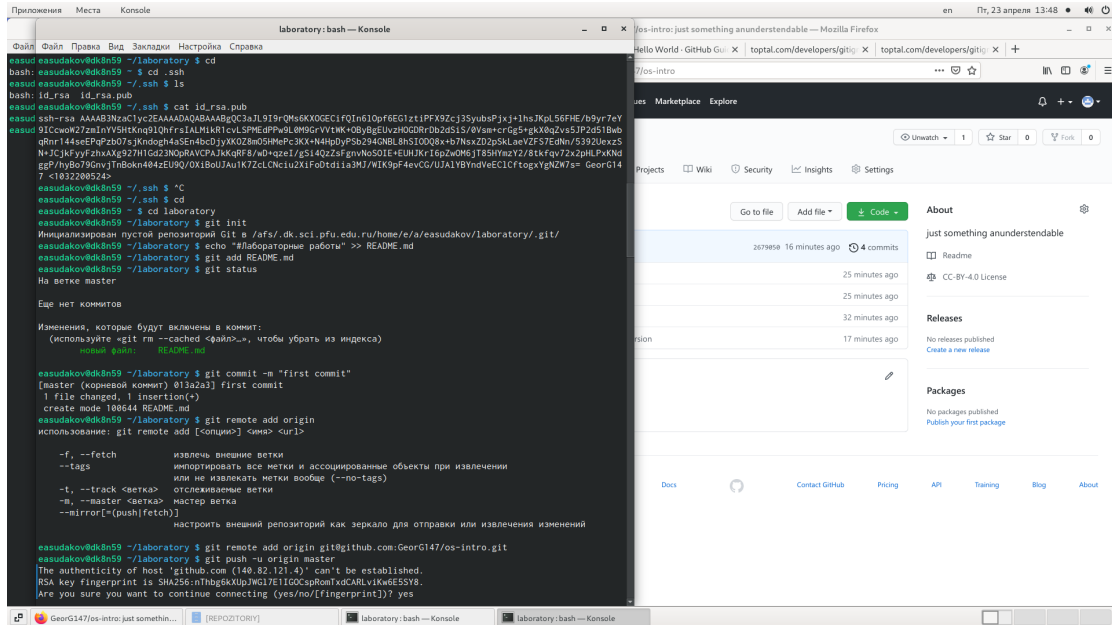
2.0.2 3. Первичная конфигурация

- Добавим файл лицензии: `wget LICENSE`
- Добавим шаблон игнорируемых файлов. Просмотрим список имеющихся шаблонов: `curl -L -s` Затем скачаем шаблон, например, для C: `curl -L -s » .gitignore` Можно это же сделать через web-интерфейс на сайте
- Выполним коммит
- Отправим на github: ### 4. Конфигурация git-flow

- Инициализируем git-flow: `git flow init` Префикс для ярлыков установим `vw`.
- Проверьте, что Вы на ветке `develop`: `git branch`
- Создадим релиз с версией `1.0.0`
- Запишем версию: `echo "1.0.0" > VERSION`
- Добавим в индекс: `git add .` `git commit -am 'chore(main): add version'`
- Зальём релизную ветку в основную ветку: `git flow release finish 1.0.0`
- Отправим данные на github: `git push --all` `git push --tags`
- Создадим релиз на github.

3 Выполнение лабораторной работы

- Создаём репозиторий егоos-intro на GitHub.
- Рабочий каталог будем обозначать как laboratory. Вначале нужно перейти в этот каталог: `cd laboratory`
- Инициализируем системы git: `git init`
- Создаём заготовку для файла README.md: `echo "# Лабораторные работы" >> README.md`
- Создаём репозиторий егоos-intro на GitHub.
- Рабочий каталог будем обозначать как laboratory. Вначале нужно перейти в этот каталог: `cd laboratory`
- Инициализируем системы git: `git init`
- Создаём заготовку для файла README.md: `echo "# Лабораторные работы" >> README.md`



The screenshot shows a terminal window on the left and a web browser on the right. The terminal window displays the following commands and output:

```
laboratory: bash — Konsole
easudakov@dk8n59 ~/$ cd
bash: easudakov@dk8n59 ~/$ cd .ssh
easudakov@dk8n59 ~/$ ssh & ls
bash: id_rsa id_rsa.pub
easudakov@dk8n59 ~/$ ssh & cat id_rsa.pub
easudakov@dk8n59 ~/$ ssh & cat id_rsa.pub
easudakov@dk8n59 ~/$ ssh & cd
easudakov@dk8n59 ~/$ cd laboratory
easudakov@dk8n59 ~/$ git init
Инициализирован пустой репозиторий Git в /afs/dk.sci.pfu.edu.ru/home/e/easudakov/laboratory/.git/
easudakov@dk8n59 ~/$ echo "# Лабораторные работы" >> README.md
easudakov@dk8n59 ~/$ git add README.md
easudakov@dk8n59 ~/$ git status
На ветке master

Еще нет коммитов

Изменения, которые будут включены в коммит:
(используйте «git rm --cached <файл>...», чтобы убрать из индекса)
    README.md

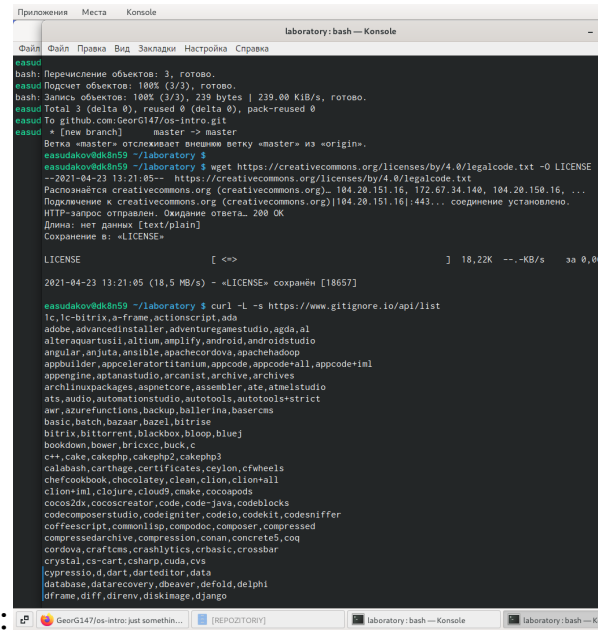
easudakov@dk8n59 ~/$ git commit -m "first commit"
[master (корневой коммит) 013a2a3] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
easudakov@dk8n59 ~/$ git remote add origin
Использование: git remote add <имя> <url>

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

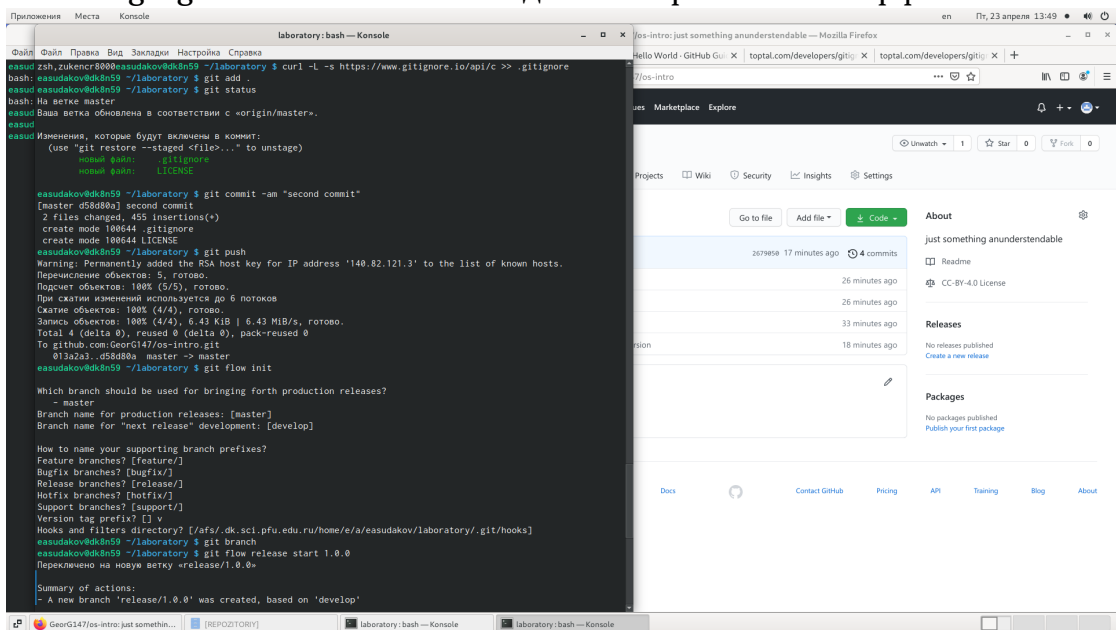
easudakov@dk8n59 ~/$ git remote add origin git@github.com:Georg147/os-intro.git
easudakov@dk8n59 ~/$ git push -u origin master
The authenticity of host 'github.com (140.82.121.4)' can't be established.
RSA key fingerprint is SHA256:n7bg8X4JwIoJ7E110CQumRiZGABRIjK6ES37B.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

The web browser on the right shows the GitHub repository page for "os-intro". The repository is just created, with 0 commits and 0 releases. The README file is visible in the repository.

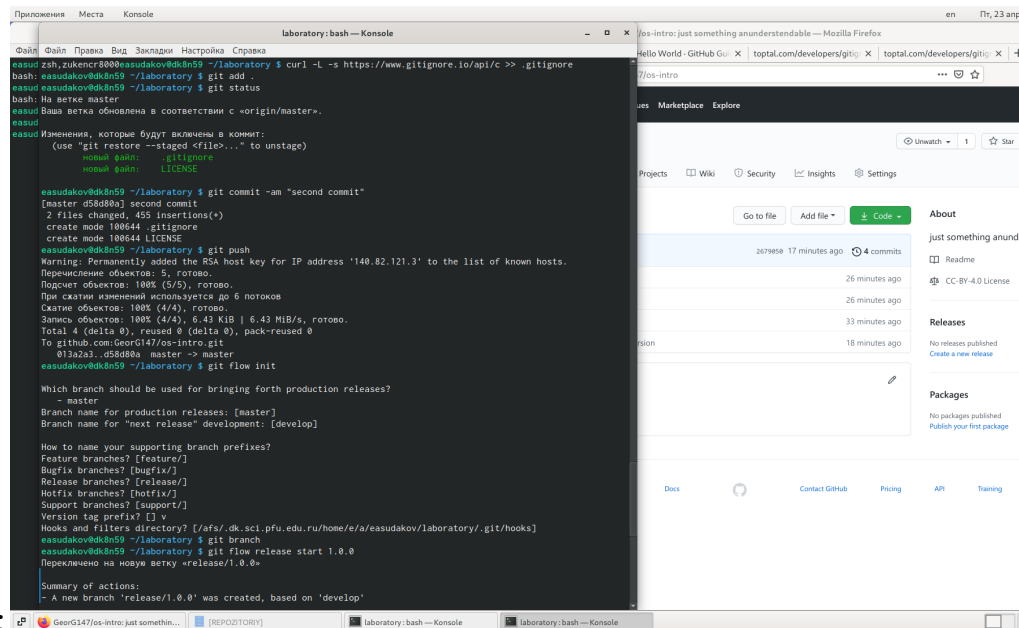
README.md



- Делаем первый коммит и выкладываем на github:
- Добавим шаблон игнорируемых файлов. Просмотрим список имеющихся шаблонов: `curl -L -s` Затем скачаем шаблон, например, для C: `curl -L -s » .gitignore` Можно это же сделать через web-интерфейс на сайте



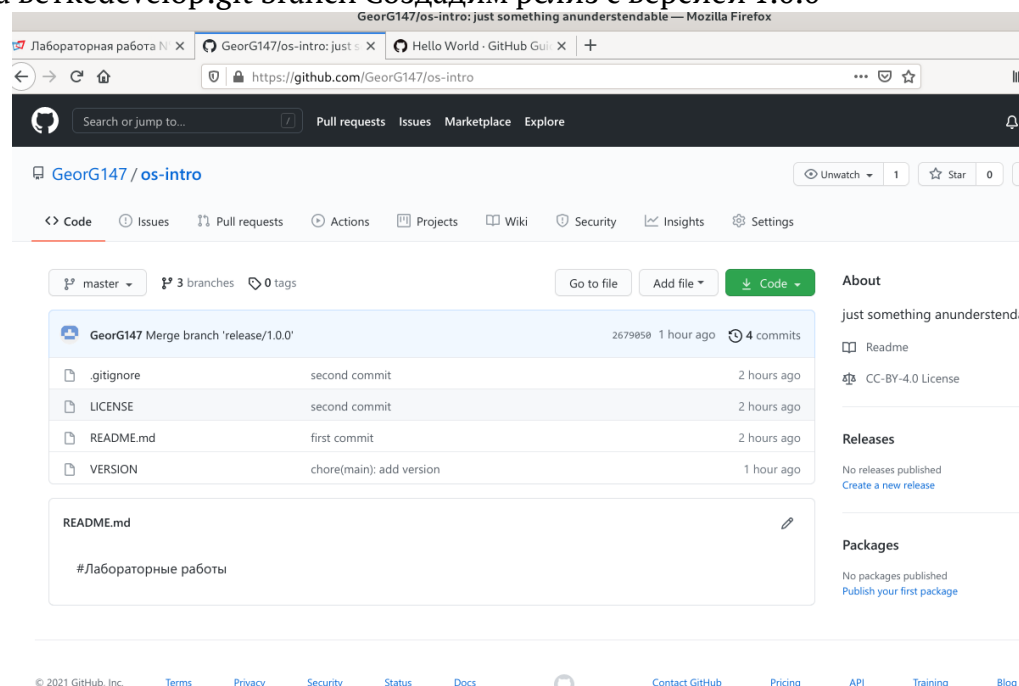
- Добавим новые файлы
- Выполним коммит



- Отправим на github:

5. Конфигурация git-flow

- Инициализируем git-flow git flow init Префикс для ярлыков установим v.
- Проверьте, что Вы на ветке develop: git branch Создадим релиз с версией 1.0.0



- Запишем версию:em

Вывод: Мы изучили как работать с репозиториями и Github