МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №1

Выполнил студент группы ИВТ-б-о-20-1
Павленко М.С. « »_____20__г.
Подпись студента_____
Работа защищена « »_____20__г.
Проверила Воронкин Р.А._____

(подпись)

Ставрополь 2021

Лабораторная работа №1

Исследование основных возможностей Git и Github

Цель работы: исследовать базовые возможности системы контроля версий Git и веб-сервиса GitHub.

Ссылка на репозиторий: https://github.com/0Roni0/1-lab.git

Порядок выполнения работы:

1. Создан общедоступный репозиторий на GitHub, в котором использована лицензия MIT, язык Python и файл .gitignore для этого языка.

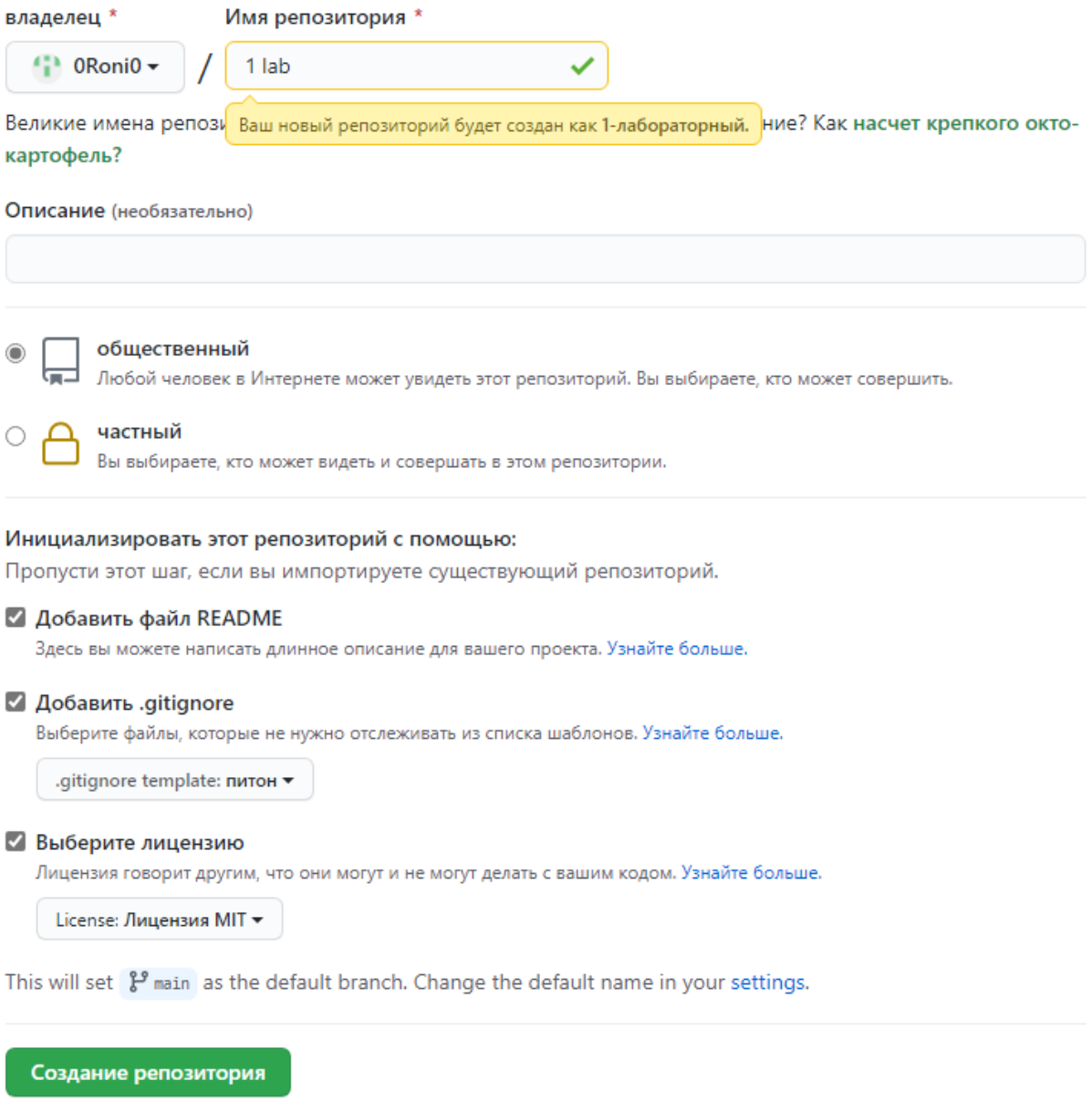

Рисунок 1. Создание общедоступного репозитория

2. Выполним клонирование репозитория на рабочий компьютер. Для этого

используем команду git clone <url>.

```
C:\Users\79624>git clone https://github.com/0Roni0/1-lab.git
Cloning into '1-lab'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

C:\Users\79624>
```

Рисунок 2. Клонирование репозитория

3. Внес изменения в репозиторий

```
79624@DESKTOP-BUV2U6A MINGW64 ~/1-lab (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

79624@DESKTOP-BUV2U6A MINGW64 ~/1-lab (main)
```

Рисунок 3. Выполнение коммита

```
79624@DESKTOP-BUV2U6A MINGW64 ~/1-lab (main)
$ git add programma.txt
```

```
79624@DESKTOP-BUV2U6A MINGW64 ~/1-lab (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md
        new file:   programma.txt
```

4. Выполнена отправка изменений в локальном репозитории в удаленный репозиторий Github командой git push

```
9624@DESKTOP-BUV2U6A MINGW64 ~/1-lab (main)
 git status
n branch main
our branch is up to date with 'origin/main'.

hanges to be committed:
 (use "git restore --staged <file>..." to unstage)
       modified:   README.md
       new file:   programma.txt


9624@DESKTOP-BUV2U6A MINGW64 ~/1-lab (main)
 git commit -m "Prog"
main f5ac7f0] Prog
2 files changed, 1 insertion(+), 1 deletion(-)
create mode 100644 programma.txt

9624@DESKTOP-BUV2U6A MINGW64 ~/1-lab (main)
 git push
numerating objects: 6, done.
ounting objects: 100% (6/6), done.
elta compression using up to 16 threads
ompressing objects: 100% (2/2), done.
riting objects: 100% (4/4), 373 bytes | 373.00 KiB/s, done.
otal 4 (delta 0), reused 0 (delta 0), pack-reused 0
o https://github.com/0Roni0/1-lab.git
  a48f42c..f5ac7f0  main -> main

9624@DESKTOP-BUV2U6A MINGW64 ~/1-lab (main)
```

Рисунок 4. Push

5. Выполнен контроль изменений на удалённом репозитории. Файлы добавлены.

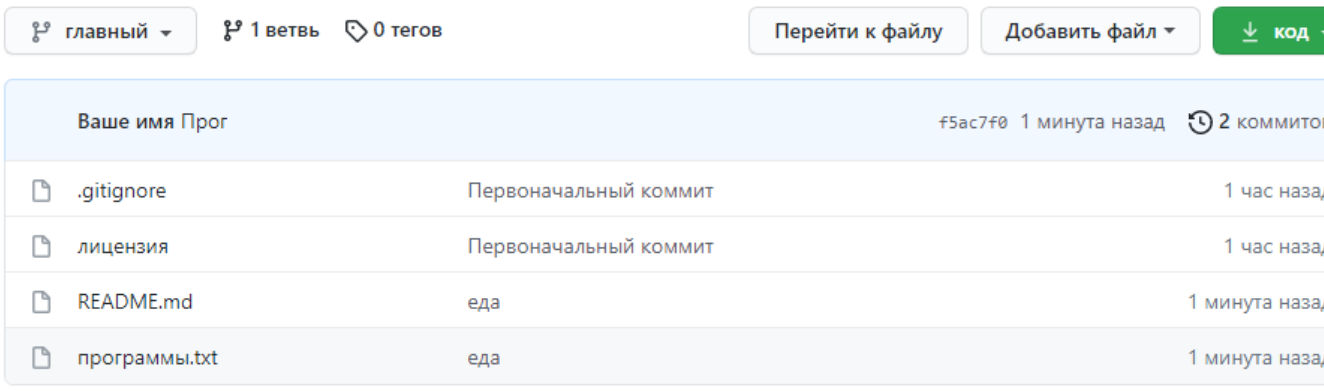

Рисунок 5. Удаленный репозиторий

Контрольные вопросы:

1. СКВ или система контроля версий – это система, регистрирующая

изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.

2. Недостатки локальных СКВ заключаются в возможности появления ошибок, изменения не тех файлов и создания общей путаницы в файлах, проблематичности во взаимодействии с другими разработчиками.Недостатки централизованных СКВ заключаются в наличии единой точкиотказа, представленной центральным сервером. При его выходе из строя невозможен контроль изменения, а при выходе из строя жесткого диска есть шанс потерять весь проект.

3. К какой СКВ относится Git?

Git относится к распределенным системам, т.к. не зависит от центрального сервера.

4. В чем концептуальное отличие Git от других СКВ?

Git не хранит и не обрабатывает данные таким же способом как другие СКВ. При каждом коммите система запоминает, как выглядит каждый файл в этот момент, и сохраняет ссылку на этот снимок. Следует, что Git эффективен в хранении бэкапов, поэтому известно мало случаев, когда кто-то терял данные при его использовании.

5. Как обеспечивается целостность хранимых данных в Git?

В Git для всего вычисляется хеш-сумма, и только потом происходит сохранение. В дальнейшем обращение к сохранённым объектам происходит по этой хеш-сумме. Это значит, что невозможно изменить содержимое файла или директории так, чтобы Git не узнал об этом.

6. В каких состояниях могут находится файлы в Git? Как связаны эти состояния?

Зафиксированный файл – файл уже сохранён в вашей локальной базе.

Измененный файл – файл, который поменялся, но ещё не был зафиксирован.

Подготовленный файл — это изменённый файл, отмеченный для включения в следующий коммит.

7. Что такое профиль пользователя в GitHub?

Профиль – публичная страница на GitHub, где есть возможность

посмотреть общедоступные репозитории проектов владельца профиля.

8. Какие бывают репозитории в GitHub?

Локальный репозиторий — это подкаталог .git, создаётся (в пустом виде) командой git init и (в непустом виде с немедленным копированием содержимого родительского удалённого репозитория и простановкой ссылки на родителя) командой git clone. Практически все обычные операции с системой контроля версий, такие, как коммит и слияние, производятся только с локальным репозиторием.

Удалённый доступ к репозиториям Git обеспечивается git- daemon, SSH- или HTTP-сервером. TCP-сервис git-daemon входит в дистрибутив Git и является наряду с SSH наиболее распространённым и надёжным методом доступа. Удалённый репозиторий можно только синхронизировать с локальнымкак «вверх» (push), так и «вниз» (pull).

9. Укажите основные этапы модели работы с GitHub.

Регистрация – создание репозитория – клонирование репозитория – контроль изменений.

10. Как осуществляется первоначальная настройка Git после установки?

После установки Git осуществляется контроль его версией (командой git version), переход в папку с локальным репозиторием (cd), связь локального и удалённого репозитория командами:

git config --global user.name <YOUR_NAME> git config --global user.email <EMAIL>

11. Опишите этапы создания репозитория в GitHub.

- нажать на кнопку «+» в правом верхнем углу страницы GitHub;
- создание имени репозитория, которое должно быть уникальным для пользователя;
- создание описания репозитория;
- выбор режима приватности репозитория (общедоступный или приватный);

- выбор, добавлять ли файлы license и .gitignore.

12. Какие типы лицензий поддерживаются GitHub при создании репозитория?

Microsoft Reciprocal License, The Code Project Open License (CPOL), The Common Development and Distribution License (CDDL), The Microsoft Public License (Ms-PL), The Mozilla Public License 1.1 (MPL 1.1), The Common Public License Version 1.0 (CPL), The Eclipse Public License 1.0, The MIT License, The BSD License, The Apache License, Version 2.0, The Creative Commons Attribution- ShareAlike 2.5 License, The zlib/libpng License, A Public Domain dedication, The Creative Commons Attribution 3.0 Unported License, The Creative Commons Attribution-Share Alike 3.0 Unported License, The Creative Commons Attribution- NoDerivatives 3.0 Unported, The GNU Lesser General Public License (LGPLv3), The GNU General Public License (GPLv3).

13. Как осуществляется клонирование репозитория GitHub? Зачем нужно клонировать репозиторий?

Клонирование репозитория нужно для работы на рабочей станции с наличием доступа ко всем его файлам.

Осуществляется при помощи команды git clone <url>. URL-адрес находится в разделе репозитория на GitHub «Code».

14. Как проверить состояние локального репозитория Git?

Для проверки состояния локального репозитория нужно ввести в терминал команду git status.

15. Как изменяется состояние локального репозитория Git после выполнения следующих операций: добавления/изменения файла в локальный репозиторий Git; добавления нового/измененного файла под версионный контроль с помощью команды git add ; фиксации (коммита) изменений с помощью команды git commit и отправки изменений на сервер с помощьюкоманды git push ?

Будет произведена загрузка/обновление файлов в удалённом репозитории.

16. У Вас имеется репозиторий на GitHub и два рабочих компьютера, с помощью которых Вы можете осуществлять работу над некоторым проектом с использованием этого репозитория. Опишите последовательность команд, с помощью которых оба локальных репозитория, связанных с репозиторием GitHub будут находиться в синхронизированном состоянии. Примечание: описание необходимо начать с команды git clone .

1) Производится клонирование репозитория (git clone) на каждый из компьютеров.

2) Для синхронизации изменений используется команду git pull.

17. GitHub является не единственным сервисом, работающим с Git. Какие сервисы еще Вам известны? Приведите сравнительный анализ одного из таких сервисов с GitHub.

GitLab — альтернатива GitHub. GitLab предоставляет не только веб- сервис для совместной работы, но и программное обеспечение с открытымисходным кодом.

SourceForge — ещё одна альтернатива GitHub, сконцентрировавшаяся на Open Source. Многие дистрибутивы и приложения Linux используют SourceForge

Launchpad — платформа для совместной работы над программным обеспечением от Canonical, компании-разработчика Ubuntu. На ней размещены PPA-репозитории Ubuntu, откуда пользователи загружают приложения и обновления.

18. Интерфейс командной строки является не единственным и далеко не самым удобным способом работы с Git. Какие Вам известны программные средства с графическим интерфейсом пользователя для работы с Git? Приведитекак реализуются описанные в лабораторной работе операции Git с помощью одного из таких программных средств.

GitHub Desktop - бесплатное приложение с открытым исходным кодом, разработанное GitHub. С его помощью можно взаимодействовать с GitHub, а также с другими платформами (включая GitLab).

Fork - GUI-клиент для macOS и Windows (с бесплатным пробным периодом). В фокусе этого инструмента скорость, дружественность к пользователю и эффективность. К особенностям Fork можно отнести кнопки быстрого доступа, встроенную систему разрешения конфликтов слияния, менеджер репозитория, уведомления GitHub.

Sourcetree - бесплатный GUI Git для macOS и Windows. Его применение упрощает работу с контролем версий и позволяет сфокусироваться на действительно важных задачах.

martGit - Git-клиент для Mac, Linux и Windows. Имеет богатый функционал: CLI для Git, графическое отображение слияний и историикоммитов, SSH-клиент, Git-Flow, программу для разрешения конфликтовслияния.

Вывод о проделанной работе: проведено исследование базовых возможностей системы контроля версий Git и веб-сервиса GitHub.