

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1.1
дисциплины
«Основы кроссплатформенного программирования»

Выполнил:
Костукайло Кирилл Николаевич
1 курс, группа ИВТ-б-о-21-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Проверил:
Кафедры инфокоммуникаций, старший
преподаватель
Воронкин Р.А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

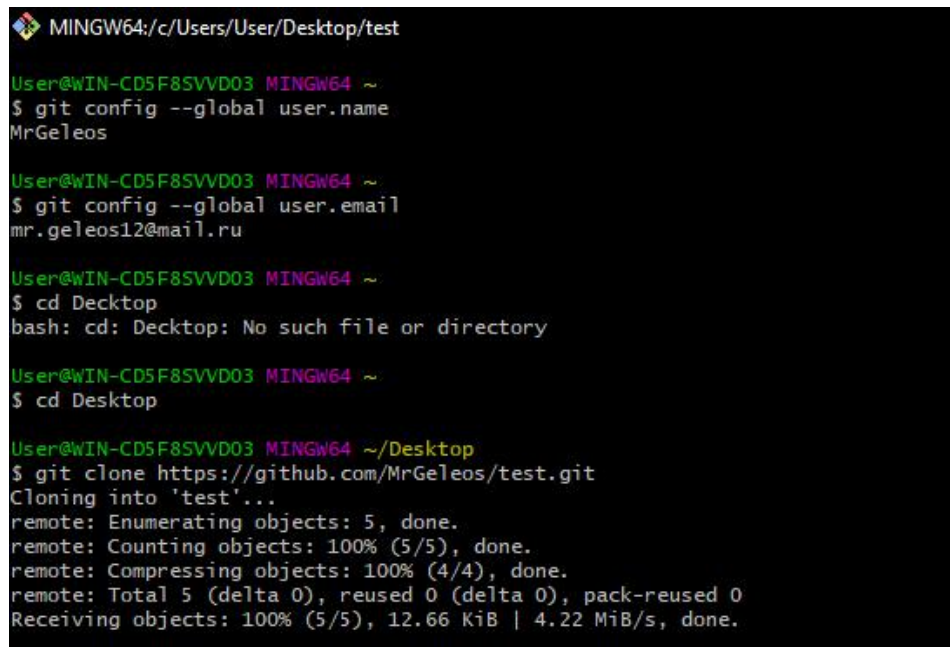
Тема: Исследование основных возможностей Git и GitHub

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

Порядок выполнения работы:

Создал общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и выбранный Вами язык программирования (выбор языка программирования будет доступен после установки флажка Add .gitignore).

Выполнил клонирование созданного репозитория на рабочий компьютер с помощью Git.



```
MINGW64:/c:/Users/User/Desktop/test

User@WIN-CD5F8SVVD03 MINGW64 ~
$ git config --global user.name
MrGeleos

User@WIN-CD5F8SVVD03 MINGW64 ~
$ git config --global user.email
mr.geleos12@mail.ru

User@WIN-CD5F8SVVD03 MINGW64 ~
$ cd Decktop
bash: cd: Decktop: No such file or directory

User@WIN-CD5F8SVVD03 MINGW64 ~
$ cd Desktop

User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop
$ git clone https://github.com/MrGeleos/test.git
Cloning into 'test'...
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), 12.66 KiB | 4.22 MiB/s, done.
```

Рисунок 1. Консоль Git

В файл README.md добавил информацию о группе и свою ФИО.

Написал небольшую программу на языке программирования C++.

Выполнив все действия, отправляем изменения в локальном репозитории в удаленный репозиторий GitHub.

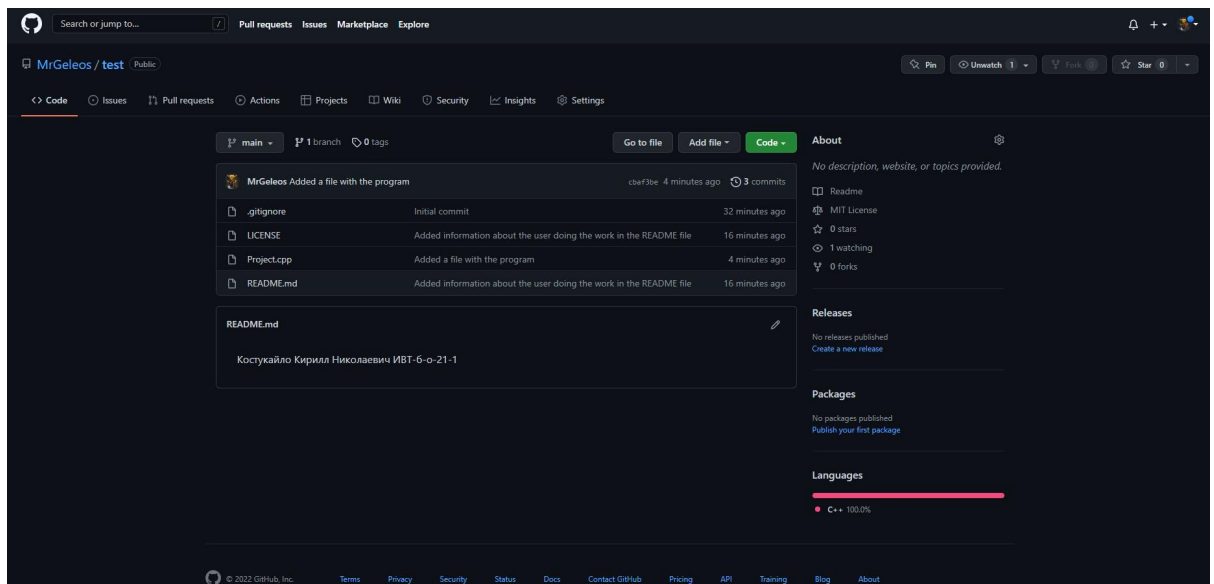


Рисунок 2.

Ответы на контрольные вопросы:

1. Что такое СКВ и каково ее назначение?

Ответ: Система контроля версий (СКВ) — это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.

2. В чем недостатки локальных и централизованных СКВ?

Ответ: многие в качестве метода контроля версий применяют копирование файлов в отдельную директорию. Такой подход очень распространён из-за его простоты, однако он невероятно сильно подвержен появлению ошибок. Можно легко забыть, в какой директории находитесь, и случайно изменить не тот файл или скопировать не те файлы, которые вы хотели.

3. К какой СКВ относится Git?

Ответ: Git относится к распределённым системам, поэтому не зависит от центрального сервера, где хранятся файлы. Git сохраняет данные в локальном репозитории.

4. В чем концептуальное отличие Git от других СКВ?

Ответ: Git не хранит и не обрабатывает данные таким же способом как другие СКВ. Каждый раз, когда вы делаете коммит, т. е. сохраняете

состояние своего проекта в Git, 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?

Ответ: 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. Данная функциональность встроена в Git на низком уровне и является неотъемлемой частью его основы. В итоге информация не теряется во время её передачи и файл не повредится без ведома Git.

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

Ответ: Зафиксированный файл, Изменённый файл, Подготовительный файл.

7. Что такое профиль пользователя в GitHub?

Профиль – ваша публичная страница на GitHub, как и в социальных сетях. Когда мы ищем работу в качестве программиста, работодатели могут посмотреть наш профиль GitHub и принять его во внимание, когда будут решать, брать нас на работу или нет.

8. Какие бывают репозитории в GitHub?

Репозиторий Git бывает локальный и удалённый.

9. Укажите основные этапы модели работы с GitHub.

Ответ: Регистрация > Создания репозитория > Клонирование репозитория.

10. Как осуществляется первоначальная настройка Git после установки?

Ответ: Убедимся, что Git установлен используя команду: `git version` > Перейдём в папку с локальным репозиторием, используя команду: `cd` > Свяжем локальный репозиторий и удалённый командами:

```
git config --global user.name <Name>
```

```
git config --global user.email <Email>
```

11. Опишите этапы создания репозитория в GitHub.

12. Какие типы лицензий поддерживаются GitHub при создании репозитория?

Ответ: 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 AttributionNoDerivatives 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?

Ответ: После создания репозитория его необходимо клонировать на ваш компьютер. Для этого на странице репозитория необходимо найти кнопку Clone или Code и щелкнуть по ней, чтобы отобразить адрес репозитория для клонирования.

14. Как проверить состояние локального репозитория Git?

Ответ: Используя 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 помощью команды `git push`?

Ответ: Файлы обновятся на 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`.

Ответ: Клонировать репозиторий на каждый из компьютеров, используя команду `git clone` и ссылку. Для синхронизации изменений используем 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.

Ответ:

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

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

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? Приведите как реализуются описанные в лабораторной работе операции Git с помощью одного из таких программных средств.

Ответ:

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

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