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

Исследование основных возможностей Git и GitHub

Выполнил студент группы

###

« »_____20__г.

Подпись студента_____

Работа защищена « »_____20__г.

Проверил доцент
Кафедры инфокоммуникаций, кандидат технических наук
Воронкин Р.А.

_____
(подпись)

Ставрополь, 2020

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

Ссылка не репозиторий https://github.com/stotubalina/lab.1

Выполнение работы:

Задание 1. Создание аккаунта на GitHub

1) Открывается сайт Github.com
2) Создаётся аккаунт

Задание 2. Изучение репозитория

1) Выполняется поиск репозитория intro-to-github
2) В файлах репозитория открывается файл README.md

Задание 3. Установка Git и создание репозитория GitHub

1) Скачивается и устанавливается Git
2) Вводим в командную строку имя и e-mail пользователя
3) Создается репозиторий с именем «lab.1»
3) Далее производится клонирование репозитория на компьютер
4) В командной строке вводится git status
5) Редактируется файл README и добавляется в репозиторий через командную строку
6) Выполняется команда git push и git pull
7) Создается программа и добавляется на репозиторий, а также редактируется файл README. Все это сопровождается 7 коммитами.
8) Добавляется в файл README описание проделанной работы в Git

Вывод по лабораторной работе:

В ходе лабораторной работы было установлено, что локальные системы контроля версий, такие как git очень удобны не только в работе на разных рабочих местах, но и для использования его как портфолио при нахождении клиента или работы как программист, так как через репозиторий легче вести

портфолио, это связано с тем, что не нужно выгружать в портфолио файлы отдельно после создания, а достаточно с помощью git выгрузить их в репозиторий.

Контрольные вопросы

1. Что такое СКВ и каково ее назначение?

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. Программисты обычно помещают в систему контроля версий исходные коды программ, но на самом деле под версионный контроль можно поместить файлы практически любого типа.

СКВ даёт возможность возвращать отдельные файлы к прежнему виду, возвращать к прежнему состоянию весь проект, просматривать происходящие со временем изменения, определять, кто последним вносил изменения во внезапно переставший работать модуль, кто и когда внёс в код какую-то ошибку.

2. В чем недостатки локальных и централизованных СКВ?

Недостаток локальных СКВ в том, что она сильно подвержен появлению ошибок. В централизованных - Централизованный сервер является уязвимым местом всей системы. Если же повреждается диск с центральной базой данных и нет резервной копии, вы теряете абсолютно всё - всю историю проекта, разве что за исключением нескольких рабочих версий, сохранившихся на рабочих машинах пользователей. Локальные системы управления версиями подвержены той же проблеме: если вся история проекта хранится в одном месте, вы рискуете потерять всё.

3. К какой СКВ относится Git?

Git относится к Распределенным системам контроля версий, так как именно такой вид СКВ дает возможность полностью копировать репозиторий.

4. В чем концептуальное отличие Git от других СКВ?

Большинство других систем хранят информацию в виде списка изменений в файлах. Эти системы представляют хранимую информацию в виде набора файлов и изменений, сделанных в каждом файле, по времени. Git не хранит и не обрабатывает данные таким способом. Вместо этого, подход Git к хранению данных больше похож на набор снимков миниатюрной файловой системы.

5. Как обеспечивается целостность хранимых данных в Git?

Невозможно изменить содержимое файла или директории так, чтобы Git не узнал об этом благодаря вычислению хеш-суммы SHA-1 хеш.

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

Зафиксированный значит, что файл уже сохранён в вашей локальной базе.

К изменённым относятся файлы, которые поменялись, но ещё не были зафиксированы.

Подготовленные файлы — это изменённые файлы, отмеченные для включения в следующий коммит

7. Что такое профиль пользователя в GitHub?

Это публичная страница на GitHub на которой можно размещать свои репозитории и проекты. Также его средствами можно организовать портфолио.

8. Какие бывают репозитории в GitHub?

Репозитории бывают публичные и приватные, то есть публичные, доступные всем, а приватные только владельцу

9. Укажите основные этапы модели работы с GitHub.

Для начало необходимо клонировать репозиторий на компьютер. Далее совершив какие-либо изменения, запросить извлечение и перенести данные на GitHub

10. Как осуществляется первоначальная настройка Git после установки?

Для начала необходимо проверить успешность установки. Это делается командой *git version* и если установка прошла корректно, то cmd выдаст информацию о версии программы. Далее необходимо командой *git config* ввести user name и user email, связанные с аккаунтом на GitHub

11. Опишите этапы создания репозитория в GitHub.

Для начала задается имя репозиторя, потом при желании приводится его описание, далее выбирается тип репозитория (публичный непубличный) и наконец, при желании, добавление файлов. gitignor и LICENSE

12. Какие типы лицензий поддерживаются GitHub при создании репозитория?

Apache 2.0, GNU v3.0, MIT, Упрощенная BSD, BSD-3-Clause, Boost 1.0, Creative Commons Zero v1.0, Eclipse 2.0, GNU Affero v3.0, GNU v2.0, GNU v2.1, Mozilla 2.0.

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

Клонирование производится с помощью команды *git clone (ссылка)*

14. Как проверить состояние локального репозитория Git?

С помощью команды 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 ?

Команда add добавит файлы в ветку git, команда commit зафиксирует изменения, после чего команда push распространит изменения на исходный репозиторий

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 .

Для начала производится клонирование репозитория на оба компьютера командой git clone, после чего репозиторий настраивается, то есть добавляется файл Readme, комитится и первичной командой push –set-upstream пушится на исходный репозиторий. Это делается для проверка работоспособности локального репозитория.

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

GitHub, Codebase, SourceForge, SourceHut, Gitorious, Bitbucket, GitLab.

Если пользоваться веб-сервисом GitLab, то для следующих задач:

- Закрытый контент команд, компаний, организаций.
- Личные рабочие репозитории (для хранения «на всякий случай»).
- Персонализированные данные.
- Проекты учебных материалов.
- Веб-разработка (удобный инструмент – GitLab Pages).

Если вы хотите, чтобы код был доступен другим, чтобы им пользовались, чтобы на него ссылались, используйте GitHub. Ссылки на репозитории GitHub вызывают большее доверие и будут просматриваться чаще. Публичная активность и проекты составят портфолио, которое может повлиять на трудоустройство.

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.

1. GitHub Desktop
2. Fork
3. Tower
4. Git Cola

5. Visual studio code

В VS Code можно клонировать репозиторий и совершить команду pull или push внутренними командами программы.