Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №1.2**

**дисциплины «Информационные технологии и программирование»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнила:  Мурашко Анастасия Юрьевна  1 курс, группа ИТС-б-0-22-1,  11.03.02 «Инфокоммуникационные технологии и системы связи», направленность (профиль) «Инфокоммуникационные системы и сети», очная форма обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Руководитель практики:  Братченко Н.Ю., канд. физ.-мат. наук, доцент, доцент кафедры инфокоммуникаций  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2023 г.

Тема: Исследование основных возможностей Gitи GitHub.

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

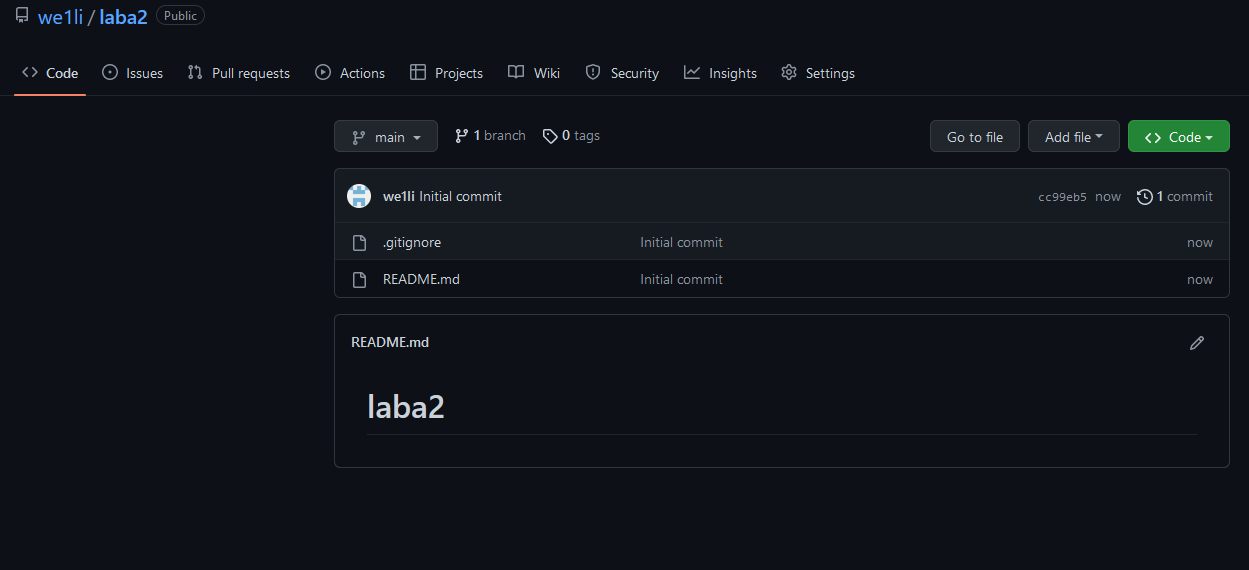
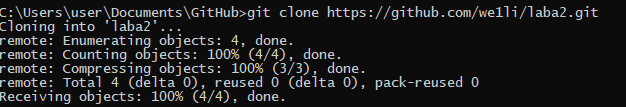
Конспект теоретического материала:

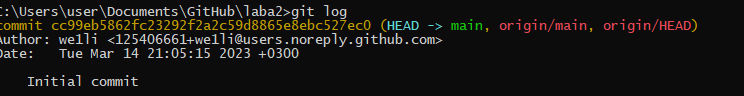
После того, как вы создали несколько коммитов или же клонировали репозиторий с уже существующей историей коммитов, вероятно Вам понадобится возможность посмотреть что было сделано — историю коммитов. Одним из основных и наиболее мощных инструментов для этого является команда git log. По умолчанию (без аргументов) git log перечисляет коммиты, сделанные в репозитории в обратном к хронологическому порядке — последние коммиты находятся вверху. Из примера можно увидеть, что данная команда перечисляет коммиты с их SHA-1 контрольными суммами, именем и электронной почтой автора, датой создания и сообщением коммита. Команда git log имеет очень большое количество опций для поиска коммитов по разным критериям. Рассмотрим наиболее популярные из них.

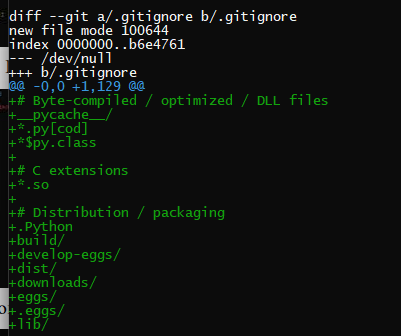
Одним из самых полезных аргументов является -p или --patch , который показывает разницу (выводит патч), внесенную в каждый коммит. Так же вы можете ограничить количество записей в выводе команды; используйте параметр -2 для вывода только двух записей.

Следующей действительно полезной опцией является --pretty . Эта опция меняет формат вывода. Существует несколько встроенных вариантов отображения. Опция oneline выводит каждый коммит в одну строку, что может быть очень удобным если вы просматриваете большое количество коммитов. К тому же, опции short , full и fuller делают вывод приблизительно в том же формате, но с меньшим или большим количеством информации.

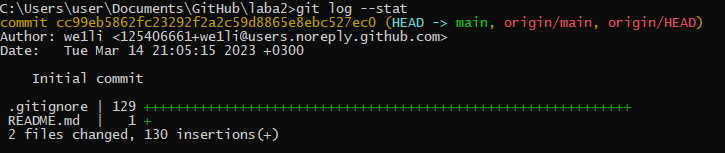
Порядок выполнения работы:

1. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и выбранный Вами язык программирования.
2. Проработайте примеры лабораторной работы. Отразите вывод на консоли при выполнении команд git в отчете для лабораторной работы.

Комманда «git clone [https://...»](https://...)

Комманда «git log».

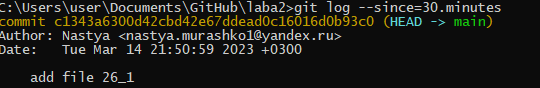
Комманда «git log -p -2»

Комманда «git log --stat»

Комманда «git log --pretty=format:"%h - %an, %ar : %s"».

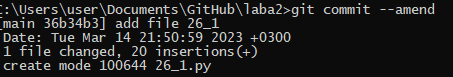


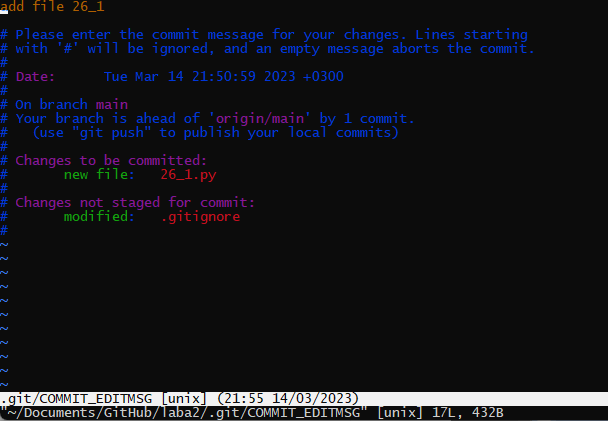
Комманда «git log --pretty=format:"%h %s" –graph».

**Ограничение вывода**

Комманда «git log --since=30.minutes».

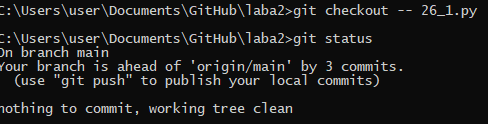
**Операции отмены**

Комманда «git commit -m "…" –amend».

****

**Отмена индексации файла**

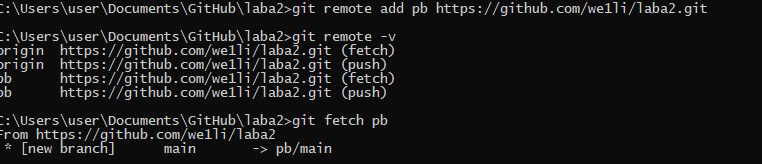
Комманда «git reset HEAD»

**Отмена изменений в файле**

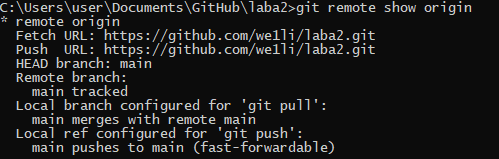
Комманда «git checkout --».

**Просмотр удалённых репозиториев**

Комманда «git remote -v».

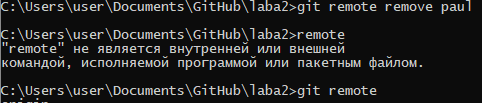
**Добавление удалённых репозиториев**

Комманда «git fetch»

**Просмотр удаленного репозитория**

Комманда «git remote show origin».

**Удаление и переименование удалённых репозиториев**

Комманда «git remote rename».

Комманда «git remote remove paul».

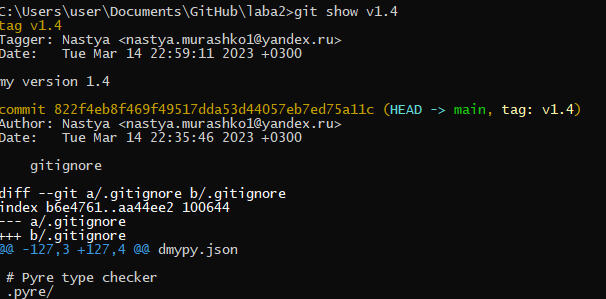
**Работа с тегами**

*Просмотр списка тегов*

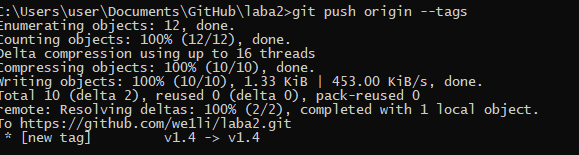
Комманда «git tag»

*Создание тегов.*

Комманда «git tag -a»

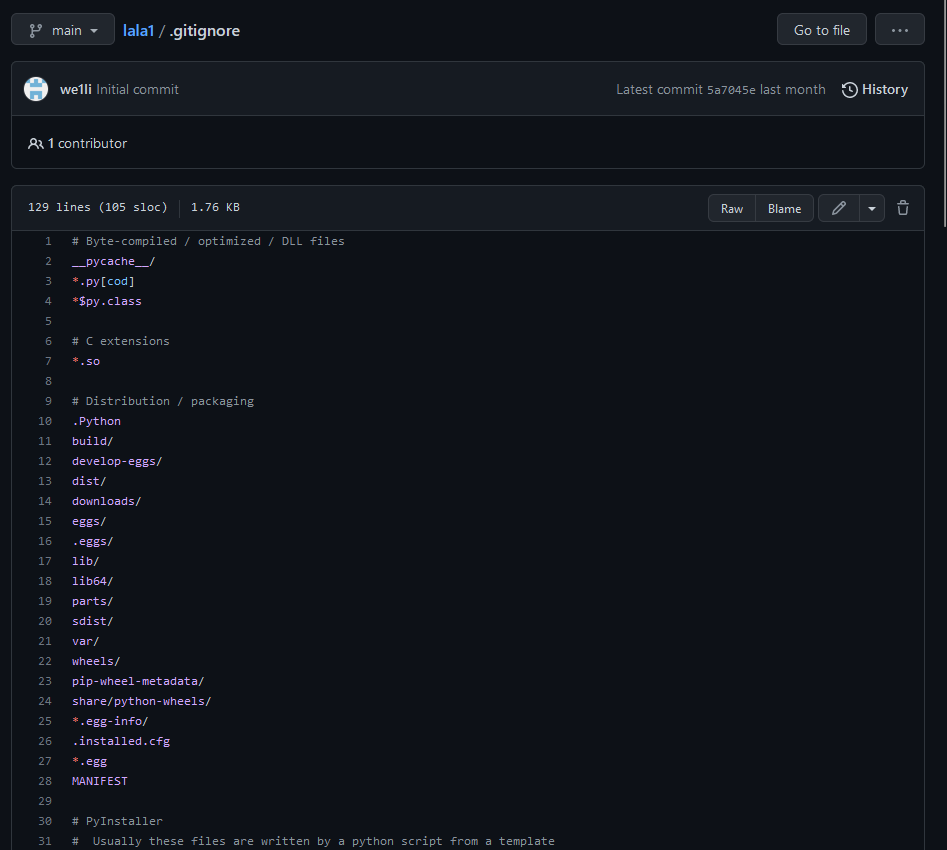
*Аннотированные теги*

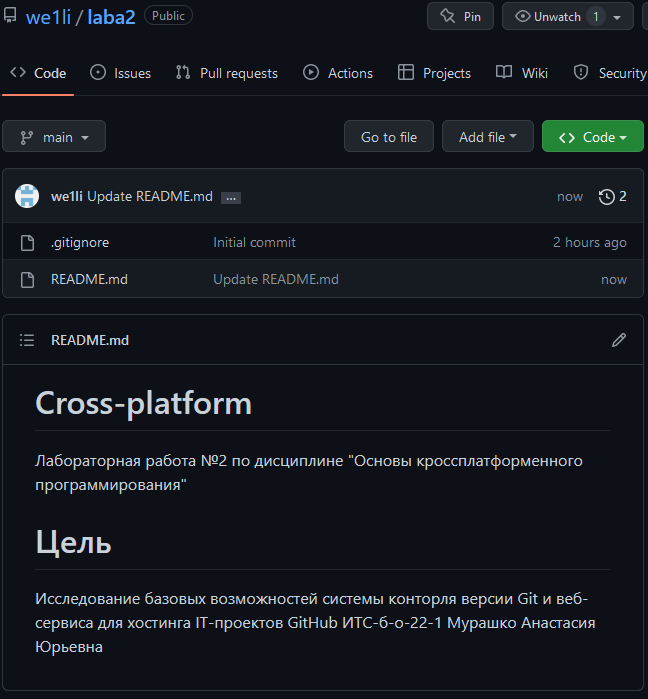
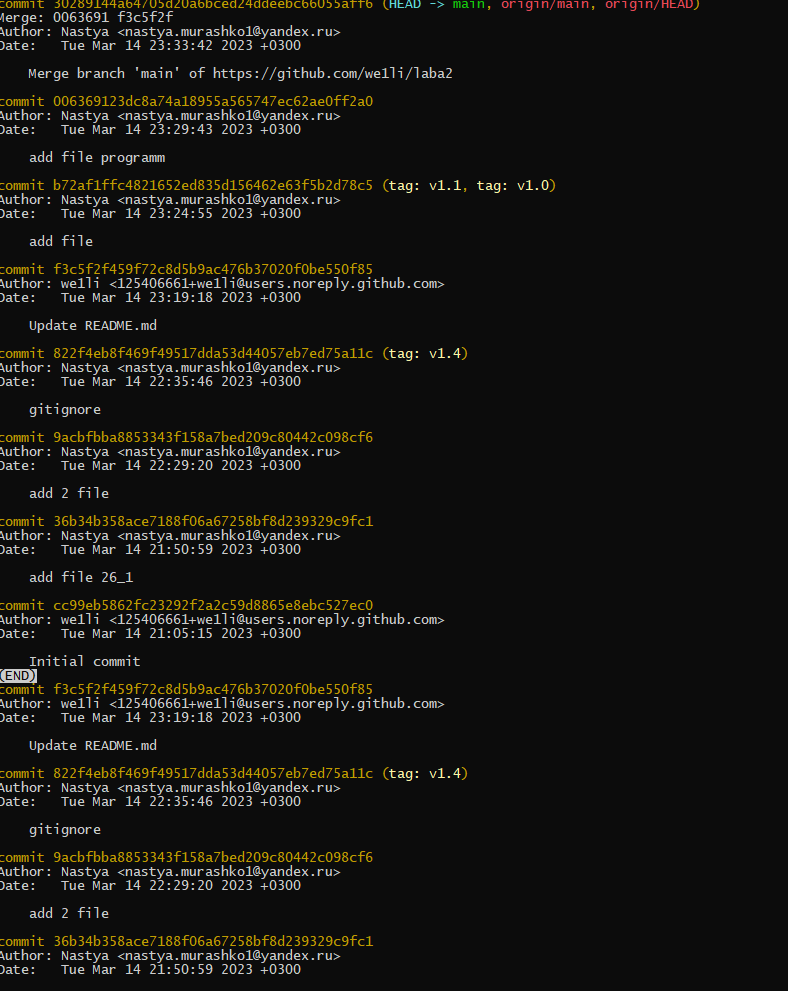
Комманда «git show v1.4».

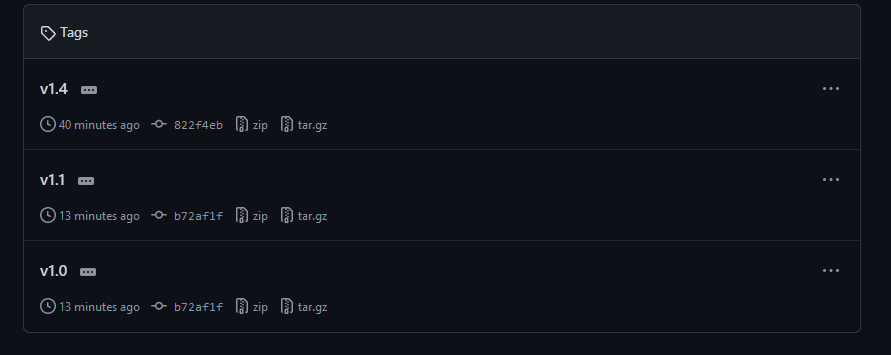
*Обмен тегами*

Комманда «git push origin –tags».

1. Дополните файл .gitignore необходимыми правилами для выбранного языка программирования и интегрированной среды разработки.



1. Добавьте в файл README.md информацию о группе и ФИО студента, выполняющего Лабораторную работу:
2. Напишите небольшую программу на выбранном Вами языке программирования. Фиксируйте изменения при написании программы в локальном репозитории. Должно быть сделано не менее 7 коммитов, отмеченных не менее 3 тэгами.



Теги коммитов.

1. Что такое СКВ и каково ее назначение?

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

1. В чем недостатки локальных и централизованных СКВ?

Локальные: возможность потери данных вследствие возникновения физических поломок оборудования; отсутствие возможности совместной разработки. Централизованные: отсутствие доступа к данным при сбое работы сервера; довольно низкая скорость работы (из-за возникновения сетевых задержек).

1. К какой СКВ относится Git?

В Git каждая рабочая копия кода сама по себе является репозиторием.

1. В чем концептуальное отличие Git от других СКВ?

Бесплатный и open-source. Можно бесплатно скачать и вносить любые изменения в исходный код; Небольшой и быстрый. Выполняет все операции локально, что увеличивает его скорость. Кроме того, Git локально сохраняет весь репозиторий в небольшой файл без потери качества данных; Простое ветвление. В других системах контроля версий создание веток— утомительная и трудоёмкая задача, так как весь код копируется в новую ветку. В Git управление ветками реализовано гораздо проще и эффективнее.

1. Как обеспечивается целостность хранимых данных в Git?

Git обеспечивает целостность хранимых данных, используя контрольные суммы в качестве идентификаторов.

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

Отслеживаемые файлы могут находится в 3 состояниях: Не изменено (Unmodified), изменено (Modified), подготовленное (Staged).

1. Что такое профиль пользователя в GitHub?

У каждого пользователя есть публичный профиль, который помогает в поиске работы. Чтобы показать свой опыт потенциальному работодателю, нужно оставить в резюме ссылку на профиль. Когда рекрутер или другой специалист перейдут по ней, то увидят информацию о вас.

1. Какие бывают репозитории в GitHub?

Репозиторий Git бывает локальный и удалённый.

1. Укажите основные этапы модели работы с GitHub.

Установка Git; добавление имени, фамилии и адреса электронной почты; ввод определенных команд для Git; загрузка изменений в состояние (staged); добавление коммита; отправка в репозиторий на сервис GitHub.

1. Как осуществляется первоначальная настройка Git после установки?

Добавление имени, фамилии и адреса электронной почты: git config --global user.name – указывает ваше имя, фамилию. git config --global user.email – указывает вашу электронную почту. git init – создает новый репозиторий Git.

11.Опишите этапы создания репозитория в GitHub.

Ввод имени для репозитория, добавление описания проекта (выборочно), выбор приватности данного репозитория, добавление дополнительных файлов, как README.md и .gitignore.

12.Какие типы лицензий поддерживаются GitHub при создании репозитория?

Academic Free License v3.0; Boost Software License 1.0; Creative Commons license family; Eclipse Public License 1.0; ISC; MIT и многие другие.

13.Как осуществляется клонирование репозитория GitHub?

Зачем нужно клонировать репозиторий? С помощью команд git clone/git push.

Чтобы упростить устранение конфликтов слияния, добавление или удаление файлов и отправку больших фиксаций.

14.Как проверить состояние локального репозитория Git? Используйте команду git status , чтобы проверить текущее состояние репозитория