Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина «Избранные главы информатики»

**ОТЧЕТ**

к лабораторной работе №1

на тему:

**«Система контроля версий Git»**

БГУИР 1-40 04 01

|  |
| --- |
| Выполнил студент группы 253505  МАХВЕЕНЯ Константин Евгеньевич |
|  |
| (дата, подпись студента) |
| Проверил доцент, кандидат технических наук  ЖВАКИНА Анна Васильевна |
|  |
| (дата, подпись преподавателя) |

Минск 2024

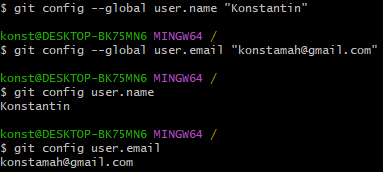
**Вариант 14.**

Задание 1.

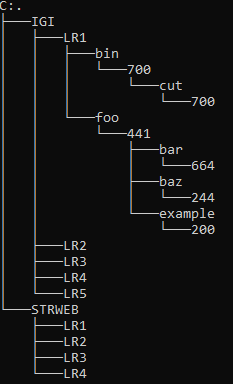
a. Проверим версию Git.



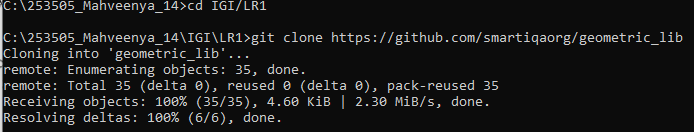
b. Выполним базовую настройку.



Задание 2. Используя консольные команды mkdir и echo создаем иерархию файлов и папок в соответствии с индивидуальным зданием. Для наглядного примера выведем составленную иерархию с помощью команды tree.

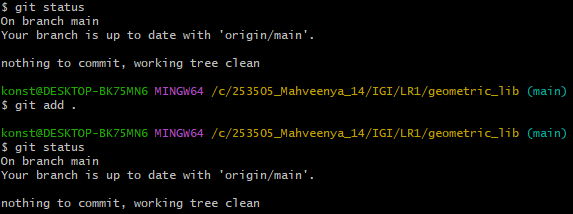


Задание 3. С помощью команды git clone выполним клонирование репозитория, заранее перейдя в нужную для этого папку.



Задание 4.

a. Изменений нет, так как не было изменено ни одного файла.

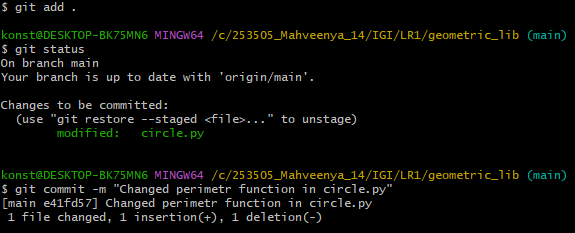


b. Изменим функцию в circle.py.

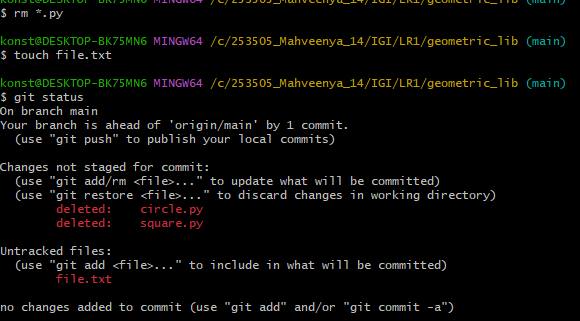




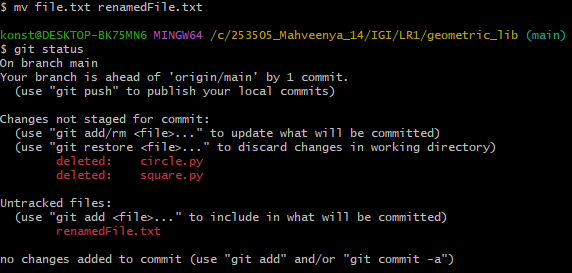
c. Проиндексировав, можно заметить, что в индексе появился файл circle.py, подготовленный к коммиту. Выполним коммит с соответствующим комментарием нему.



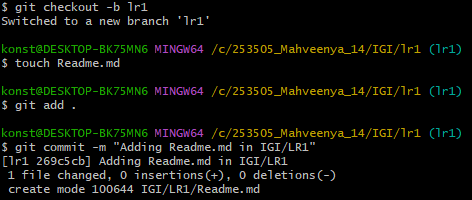
d. Добавим и удалим файлы в репозитории.



e. Переименуем файл с помощью команды mv.

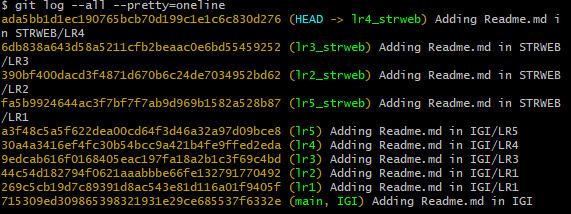


f. Создаем новую ветку, переходим на нее и создаем на ней файл, после чего фиксируем это коммитом для каждой из заданных папок.



g. Выведем историю коммитов.

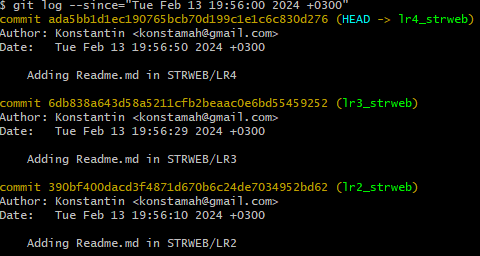
- полностью:



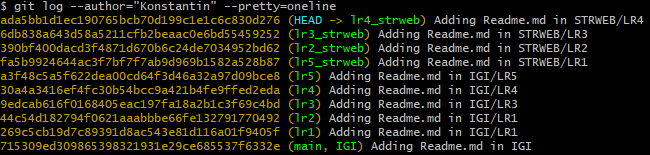
- последние 3:



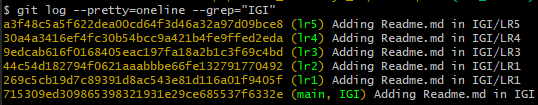
- после определенного времени:



- выполненные определенным автором:

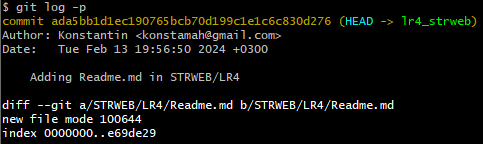


- содержащие определенную подстроку:

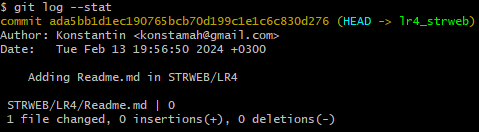


h.

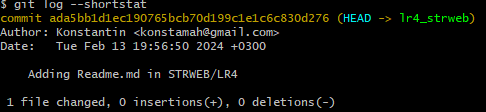
- p – показывает разницу, внесенную в коммит.



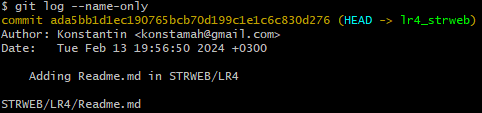
- stat – показывает ту же информацию, что и p, но содержит разницу для каждой записи.



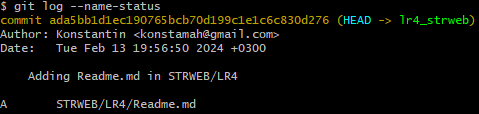
- shortstat – не содержит информацию о разнице, внесенной в коммит.



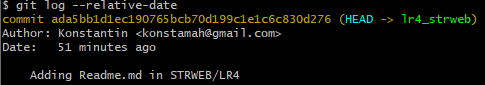
- name-only – содержит информацию о сообщении коммита.



- name-status – показывает имена и статус измененных файлов.



- relative-date – показывает, сколько времени прошло от создания коммита.



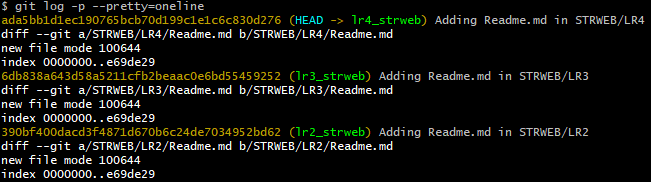
- pretty – выводит информацию о коммите в удобном режиме, например, в одну строку.



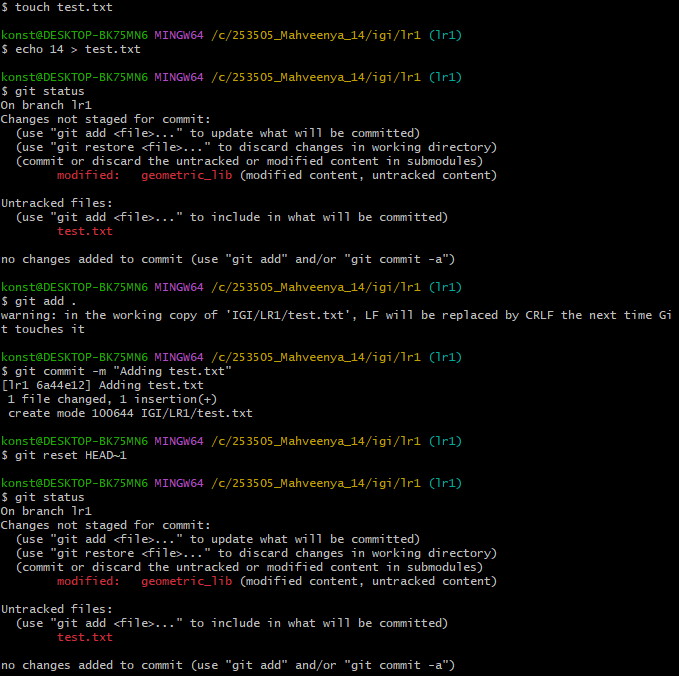
- graph – выводит информацию о коммите в иерархическом виде.



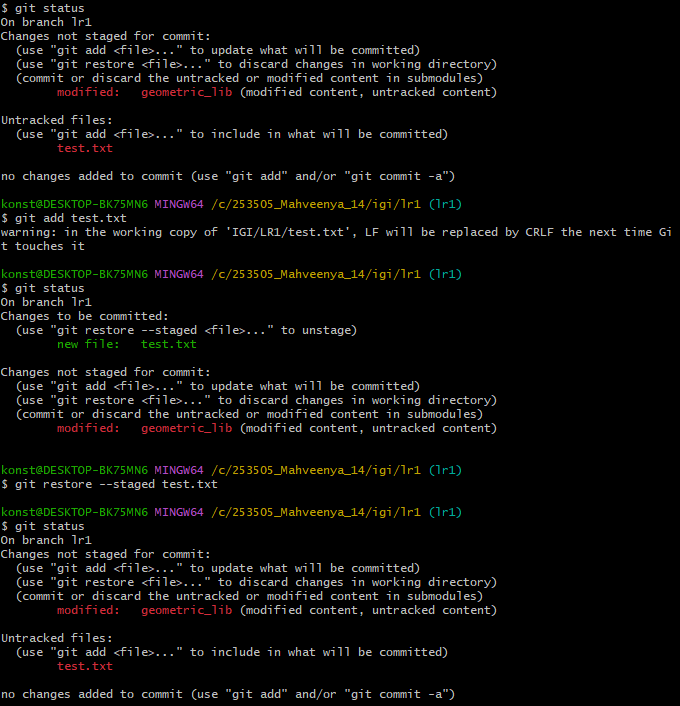
i. С помощью уже известной команды найдем разницу коммитов.



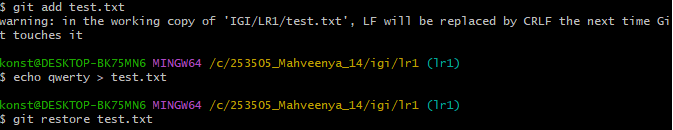
j. Создаем файл с номером варианта, добавляем в индекс и делаем коммит, после чего с помощью команды reset отменяем коммит, перенося ссылку HEAD на один коммит назад.



k. Внесем файл в область индекса, а затем отменим это действие с помощью команды git restore с ключом –staged.



l. Для восстановления файла внесем его изначальную реализацию в индекс, а затем в помощью команды git restore мы можем восстановить любой файл из индекса, следовательно, восстанавливаем test.txt.

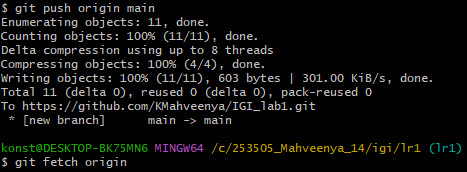


Задание 5.

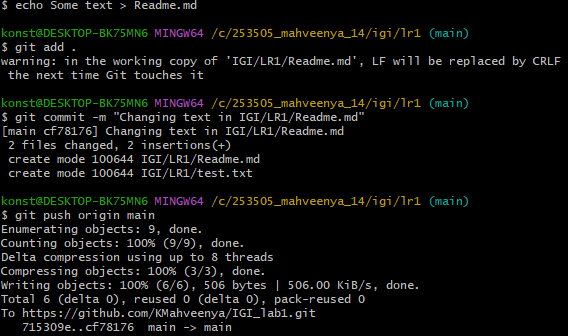
a. Подключаем удаленный репозиторий к локальному.



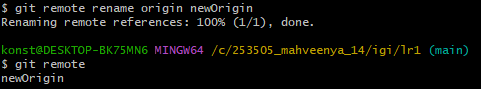
b. Отправляем изменения на удаленный репозиторий и для проверки получаем изменения из него же.



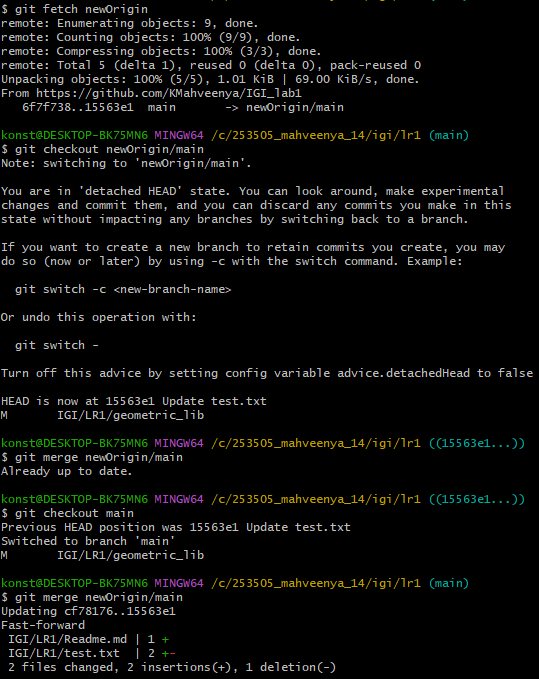
c. Переходим на ветку main и, например, изменяем текст в файле Readme.md, после чего добавляем файл в индекс, делаем коммит и добавляем изменения на удаленный репозиторий.



d. С помощью ключа rename переименовываем удаленный репозиторий.



e. Внесем изменения в репозитории на гитхаб после чего запросим изменения оттуда в локальный репозиторий. Можно заметить, что рабочая копия не была обновлена, поэтому можно слить две главные ветки в одну. Тем самым мы получим обновленную рабочую копию, соответствующую удаленному репозиторию.



f. Просмотрим историю изменений.

