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 6-05-0612-02 83

|  |
| --- |
| Выполнила студентка группы 353501  МАЛИНОВСКАЯ Евгения Александровна |
|  |
| (дата, подпись студента) |
| Проверила доцент каф. Информатики  ЖВАКИНА Анна Васильевна |
|  |
| (дата, подпись преподавателя) |

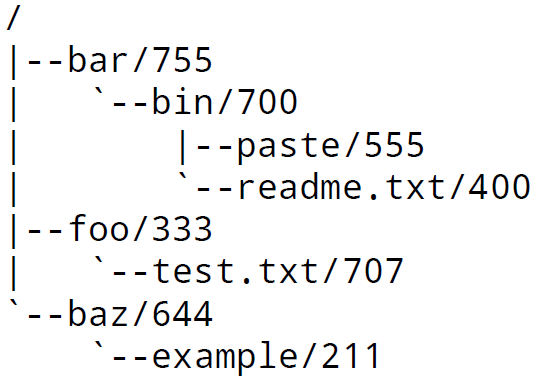
Минск 2024

**1 ЦЕЛЬ ВЫПОЛНЕНИЯ ЗАДАНИЯ**

Познакомиться с возможностями и получить практические навыки работы с системой контроля версий Git.

**2 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ**

**Вариант 11.** Создать структуру папок согласно варианту:



**3 ВЫПОЛНЕНИЕ ЗАДАНИЯ**

Для выполнения задания сначала была скачана программа GitHub Desktop для отслеживания локальных репозиториев без использования командной строки.

Далее с помощью комбинации клавиш Win+R была открыта командная строка, с помощью которой будет происходить выполнение задания. На рисунке 1 изображена командная строка cmd, через которую на диске D:\ была создана папка 353501\_MALINOVSKAYA\_11 (команда mkdir 353501\_MALINOVSKAYA\_11) и проверена на корректность создания (заходом в папку командой cd 353501\_MALINOVSKAYA\_11).

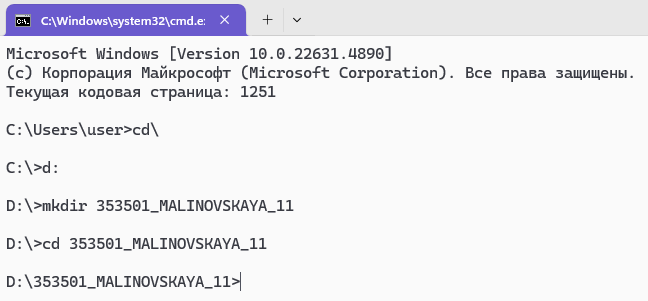


Рисунок 3.1 – Создание папки 353501\_MALINOVSKAYA\_11

В соответствии с заданием была создана система папок с помощью команды mkdir. Команда tree D:\353501\_MALINOVSKAYA\_11 помогла наглядно увидеть строение каталога в виде дерева, продемонстрированного на рисунке 3.2.

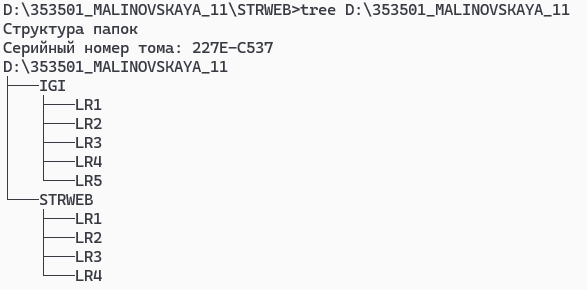


Рисунок 3.2 – Система созданных папок

Далее для папки 353501\_MALINOVSKAYA\_11 был создан локальный репозиторий с помощью команды git init, после чего состав папки изменился. Это видно на рисунке 3.3, изображающей проводник.

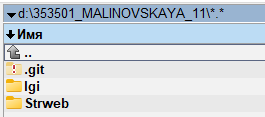
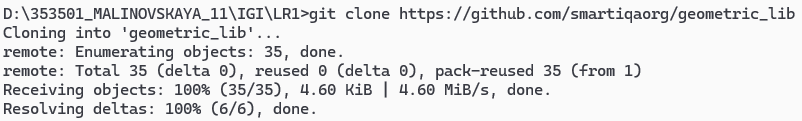


Рисунок 3.3 – Система созданных папок

В папке LR1 с помощью копирования проекта c github.com появились новые каталоги и файлы, что видно на рисунке 3.4.



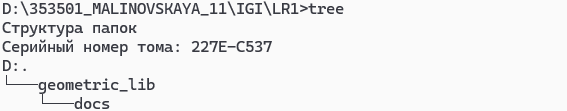
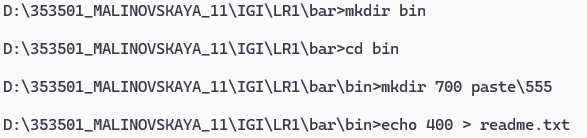


Рисунок 3.4 – Система созданных папок

В соответствие с индивидуальным заданием в LR1 были добавлены несколько папок и файлов c текстом, что продемонстрировано на рисунке 3.5.



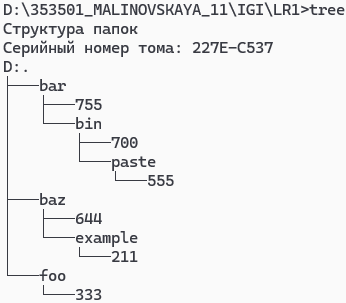
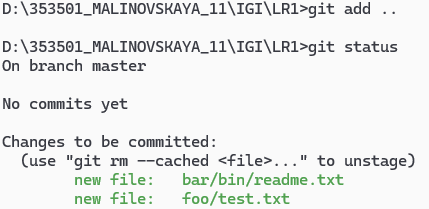


Рисунок 3.5 – Индивидуальное задание

C помощью команд git add, git status, git commit удалось успешно закоммитить изменения в репозитории с комментарием и отследить все изменения, что видно на рисунке 3.6.



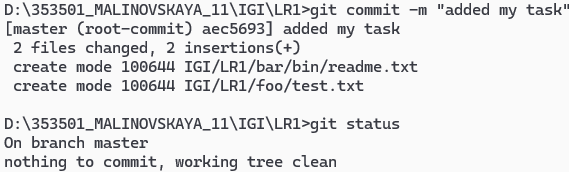


Рисунок 3.6 – Комиты к индексированным файлам

На рисунке 3.7 показано, как с помощью команды git branch были созданы несколько веток. Команда git log --all --decorate --oneline –graph вывела историю коммитов с несколькими флагами.

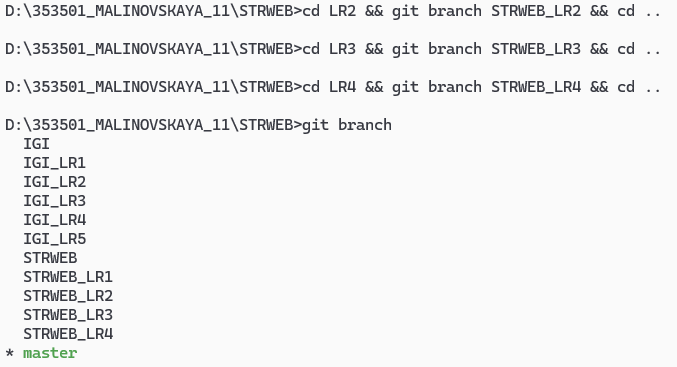




Рисунок 3.7 – Добавленные ветки

В папке LR1 был создан текстовый файл test.txt с текстом “11”. Затем был сделан коммит добавления этого файла локально и отменен, рисунок 3.8. На рисунке 3.9 показана отмена индексации файла с помощью git reset test.txt.

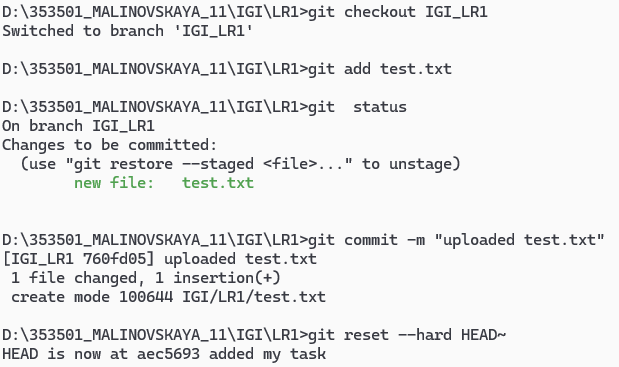


Рисунок 3.8 – Отмена коммита

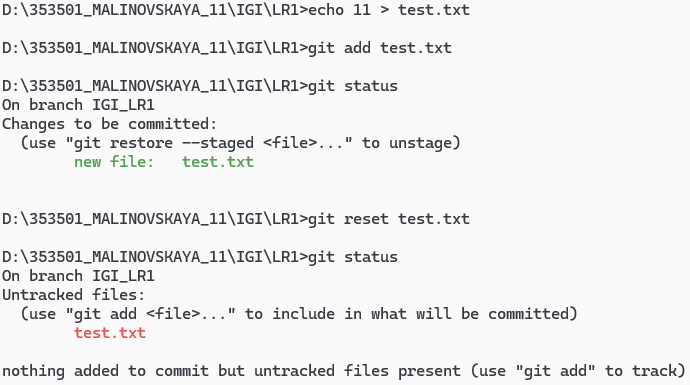


Рисунок 3.9 – Отмена индексации

С помощью команд git push origin master была запушена основная ветка и создан удаленный репозиторий, что видно на рисунке 3.10. Команда git push была использована для создания локальных и удаленных веток репозитория, рисунок 3.11.

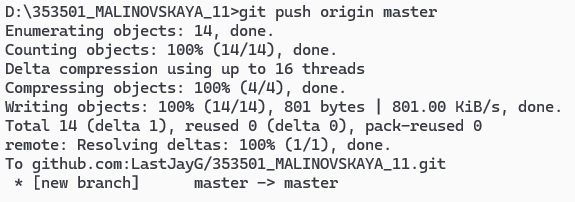


Рисунок 3.10 – Создание удаленного репозитория

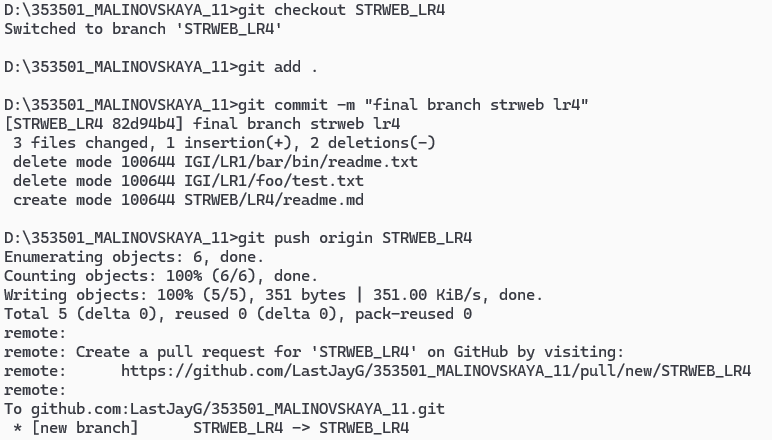


Рисунок 3.11 – Создание локальных и удаленных веток для папок LR

Для просмотра истории коммитов в репозитории была использована команда git log (рисунок 3.12), позволяя анализировать внесённые изменения, авторов и временные рамки разработки. Флаги в команде git log позволяют фильтровать и настраивать вывод коммитов: -n <number> (рисунок 3.13) ограничивает количество выводимых коммитов, --since <date> и --after <date> показывают коммиты, сделанные после указанной даты, в то время как --until <date> и --before <date> отображают коммиты до определённой даты. Флаги --author <name> (рисунок 3.14) и --committer <name> фильтруют коммиты по автору и коммиттеру соответственно, а --grep <pattern> позволяет искать коммиты по сообщениям, отображая только те, которые содержат указанный текст.



Рисунок 3.12 – Просмотр всех последних коммитов



Рисунок 3.13 – Просмотр двух последних коммитов из четырех



Рисунок 3.14 – Просмотр коммитов пользователя LastJayG

На рисунке 3.15 показано удаление и создание соответствующих папок, после чего был выполнен коммит, который затем был отменён, откатившись на назад с помощью команды git reset --hard HEAD.

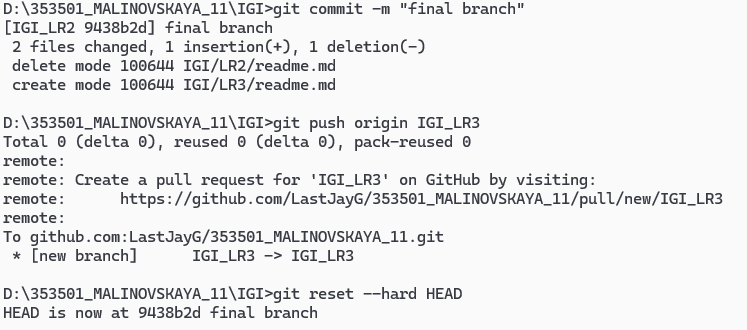


Рисунок 3.15 – Создание нового коммита и его отмена

Для нахождения разницы между коммитами сначала были определены их хэши с помощью команды git log –oneline, а затем была найдена разница между коммитами a35f171 и aec5693 с помощью команды git diff <name> <name> (рисунок 3.16).

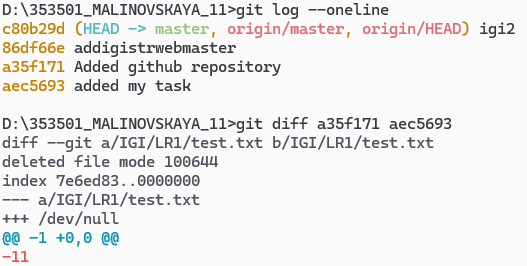


Рисунок 3.16 – Разница между коммитами a35f171 и aec5693

Для просмотра всех веток, локальных и удаленных, была использована комманда git branch –all (рисунок 3.16).

Сначаала была создана локально и удаленно лишняя ветка extrabranch, а потом с помощью команды git push origin --delete <имя ветки> была удалена удаленно, но в локальном репозитории сохранилась. Чтобы удалить ветку локально, можно воспользоваться командой git branch -D <имя ветки>.



Рисунок 3.16 – Просмотр всех веток



Рисунок 3.16 – Удаление только удаленной ветки extrabranch

**ВЫВОД**

В ходе выполнения лабораторной работы по теме "Система контроля версий Git" были изучены ключевые возможности Git. После успешной установки и настройка Git на локальном компьютере была создана структура папок для репозитория в соответствии с заданием. В процессе работы были выполнены основные команды Git, такие как git init, git add, git commit, а также изучены способы управления ветками и индексацией файлов. Это позволило наглядно увидеть, как Git помогает отслеживать изменения и управлять проектом, обеспечивая возможность работы в команде без конфликтов.

Также был проведен ряд операций с удаленным репозиторием на GitHub, включая создание и переименование репозитория, а также перенос изменений между ветками. Все действия фиксировались с помощью коммитов, что позволило строить историю изменений проекта и использовать различные флаги команды git log для глубокого анализа.