МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ

Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «Информационные технологии»

Информационные технологии и программирование

Лабораторная работа №1

Выполнила: Студент группы

БВТ2207

Кривенок Вадим

Москва

2023

**Цель работы:** изучение базовых функций и подходов к командной разработке в системе контроля версий Git.

**Задание:** определить проект (состоящий из нескольких файлов), который будет добавлен под версионный контроль, и произвести ряд операций в СКВ.

**Выполнение.**

Копирую репозиторий и инициализирую локальный репозиторий командой git init.

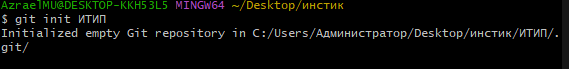


Рис.1 - инициализация локального репозитория.

Вношу ряд изменений в файл(ы) и выполняю коммиты (кол-во не меньше 2)

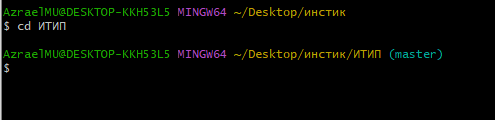
Для начала перехожу в директорию. 

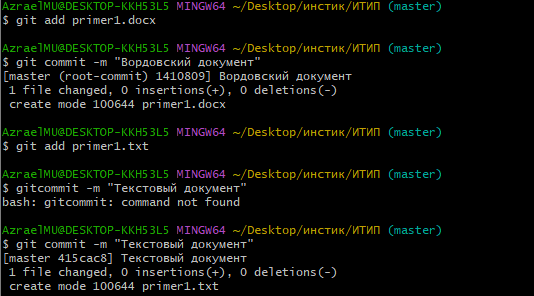
Рис.2 – переход в директорию. 

Рис.3 – добавление файлов и их коммиты.

Откатываю изменения до одного из предыдущих коммитов одним из двух способов:

1. git revert HEAD (команда, которая откатывает коммит на один назад)

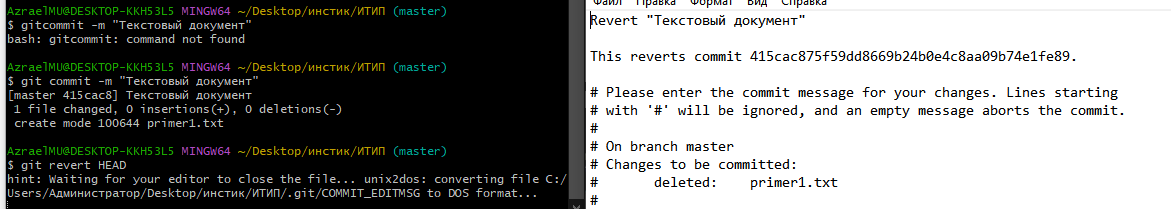


Рис.4 – откат до предыдущего коммита (1 способ)

1. git revert <Хэш коммита> (предварительно нужно посмотреть хэш коммита командой git log)

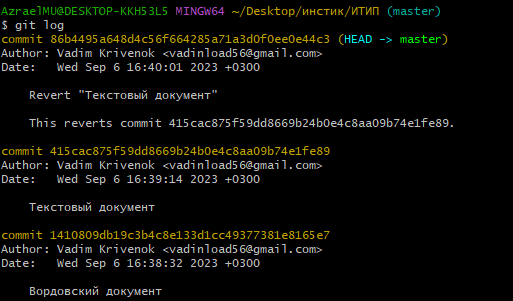


Рис.5 – команда git log и хэш коммита

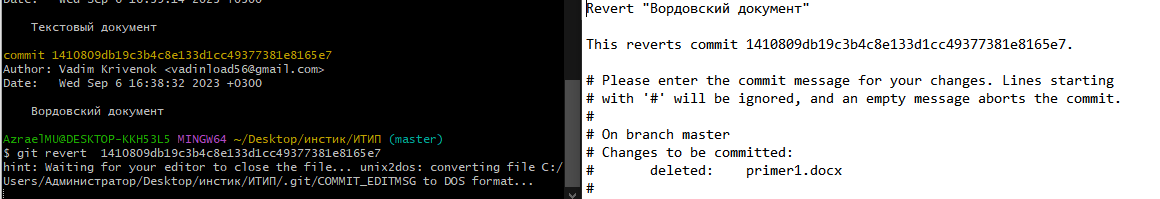


Рис.6 – откат до определенного коммита (2 способ)

Создаю, переименовываю и удаляю ветку.

Для создания ветки использую команду git branch <имя моей ветки>

Для переименования ветки использую git branch -m <изначальное имя> <измененное имя>\*можно просмотреть все ветки командой git branch

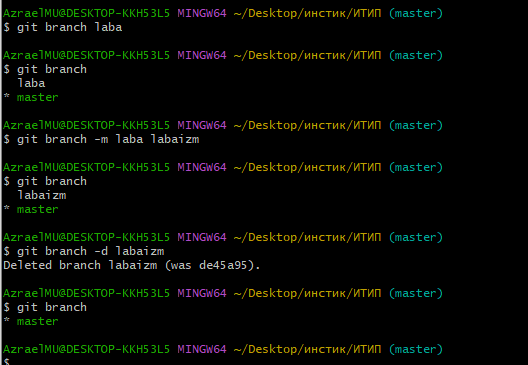
Для удаления ветки использую команду git branch -d <имя ветки>

Рис.7 – создание, переименование и удаление ветки.

Слияние ветки с веткой мастер с решением конфликта содержания файлов

Создаю новую ветку и выполняю слияние с веткой master чрез команду git merge.

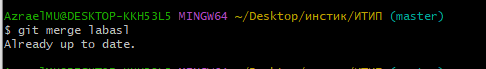


Рис.8 – слияние с веткой master.

Настройка игнорирования git’ом файлов определенного расширения с использованием glob-шаблонов

Создаю файл .gitignore в корневой папке репозитория.

Использую команду touch .gitignore

Рис.9 – создание файла .gitignore

Использую команду vim .gitignore для заполнения файла.

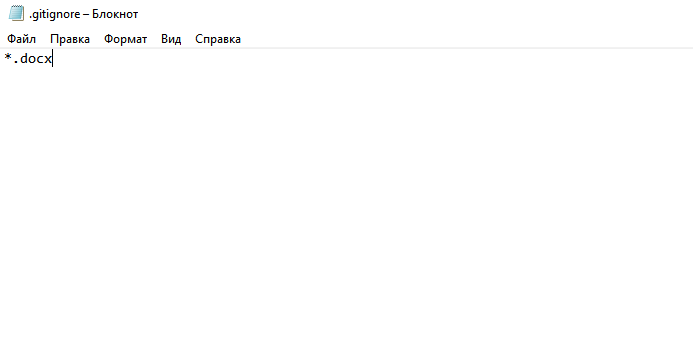
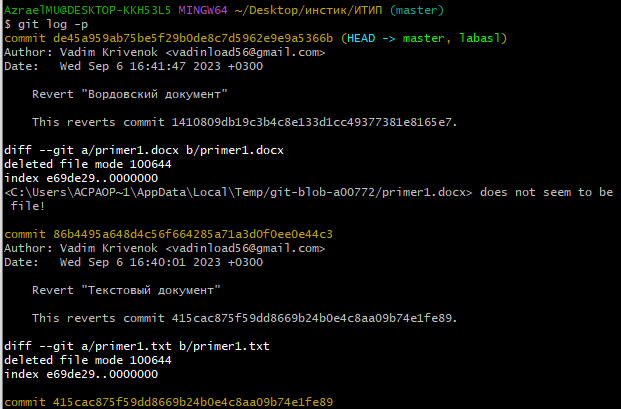


Рис.10 – файл .gitignore

Вывод лога коммитов в измененном пользовательском формате

Использую команду git log –p

Рис.11 – команда git log –p

Настройка удалённого репозитория: получение и отправка изменений в репозиторий.

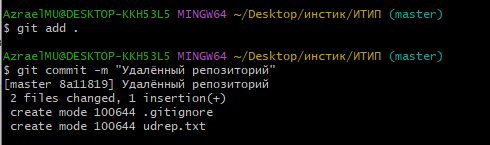
Добавлю в инициализированный репозиторий свой файл (udrep.txt) и командой git add . добавлю их в индекс для коммита. 

Рис.12 – добавление файла и коммит.

Создаю удалённый репозиторий GitHub.

Связываю между собой локальный репозиторий с удалённым, командой git remote add origin “ссылка на репозиторий GitHub

Рис.13 – связывание между собой локального репозитория с удаленным.

Выполняю пуш из локального репозитория в удаленный репозиторий GitHub командой git push --set-upstream origin main.

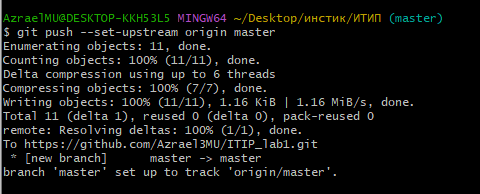


Рис.14 – пуш из локального репозитория в удаленный.

Организация совместной работы над проектом: клонирование чужого репозитория, выполнение коммитов, отправка изменений.

Копирую ссылку на удаленный репозиторий GitHub.

Перехожу в папку где он будет размещён и клонирую его в неё командой git clone “ссылка на GitHub”

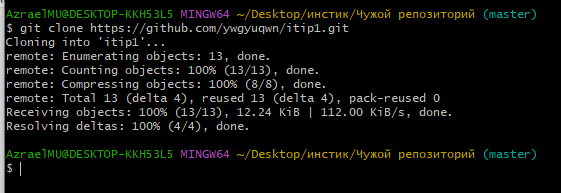


Рис.15 – клонирую чужой репозиторий.

Перехожу в удалённый репозиторий (клонированный в локальную папку).

Добавляю изменения:

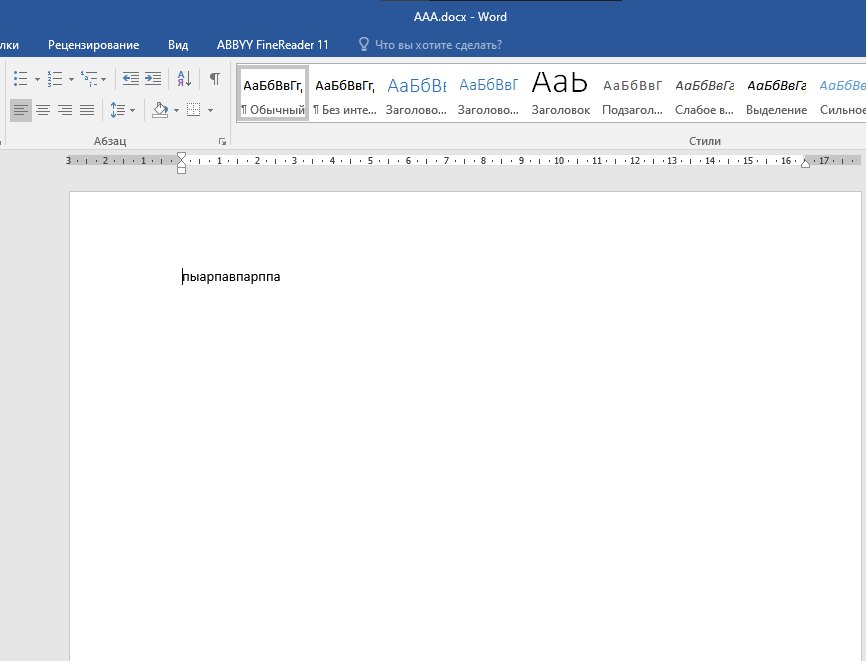


Рис.16 – изменения в чужом файле из репозитория.

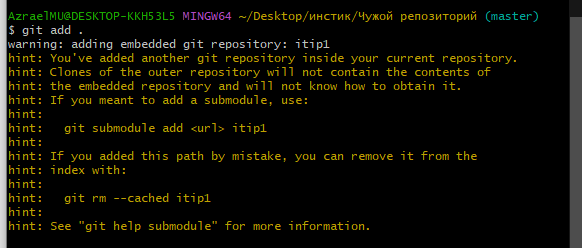


Рис.17 – выполнение команды git add . .

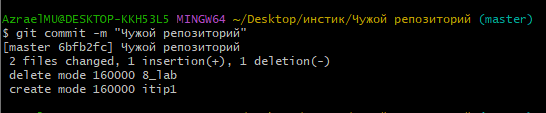
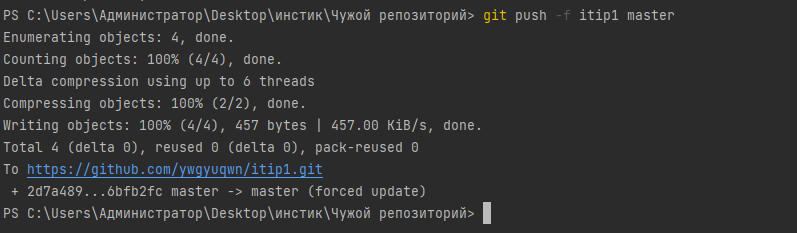
Делаю коммит командой git commit – m: 

Рис.18 – коммит.

Выполняю пуш в чужой репозиторий

 Рис.19 – пуш.