Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Ордена Трудового Красного Знамени

федеральное государственное бюджетное образовательное учреждение высшего образования

МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ

Кафедра «Математической кибернетики и информационных технологий»

Информационные технологии и программирование

Лабораторная работа по теме

«Основы работы с системой контроля версий»

Выполнил:

студент группы БВТ2202

Сальников А. Е.

**Цель работы:** изучить основные возможности, получить навыки использования распределённой системы управления версиями git.

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

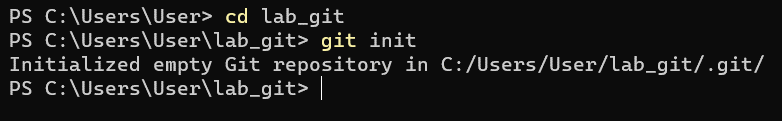
**Создание репозитория или клонирование существующего**

Внесение ряда изменений в файл(ы) и выполнение коммитов (кол-во не меньше 2)

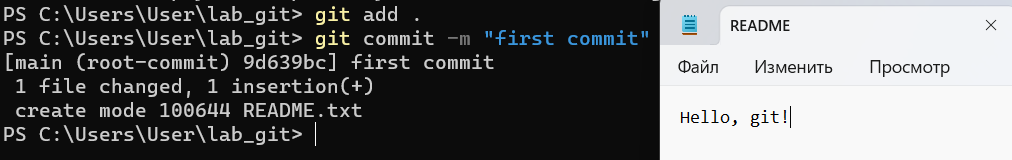
1. Откат изменений до одного из предыдущих коммитов двумя разными способами.
2. Создание, переименование и удаление ветки.
3. Слияние ветки с веткой мастер с решением конфликта содержания файлов.
4. Настройка игнорирования git’ом файлов определенного расширения с использованием glob-шаблонов.
5. Вывод лога коммитов в измененном пользовательском формате.
6. Настройка удалённого репозитория: получение и отправка изменений в репозиторий.
7. Организация совместной работы над проектом: клонирование чужого репозитория, выполнение коммитов, отправка изменений.

**Ход работы.**

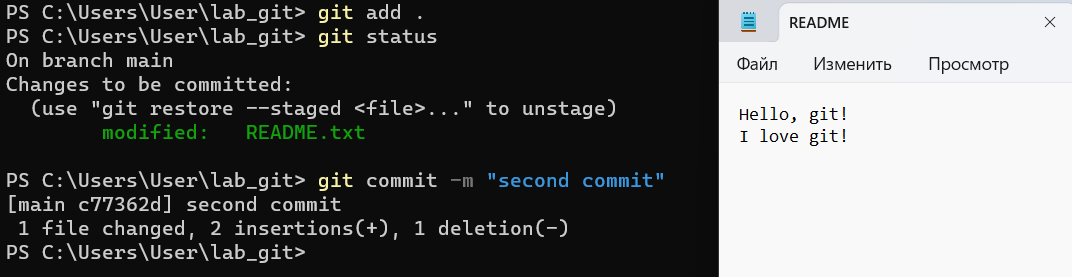
Первым делом инициализируем репозиторий командой "git init".



Создадим текстовый файл и внесём туда какой-то текст. Добавляем файл в индекс гита и коммитим.



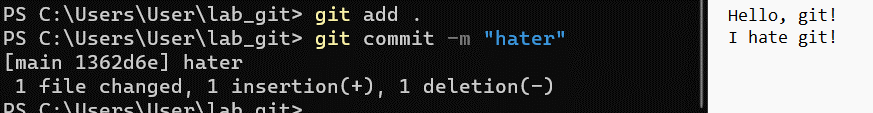
Изменим содержимое файла и повторим вышеуказанные действия.



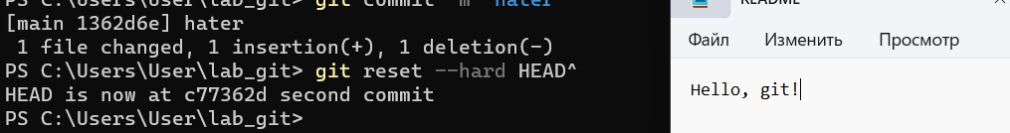
Теперь, например мы хотим вернуться к состоянию первого коммита, воспользуемся командой "git revert". Она возвращает нас в прошлое состояние, не удаляя коммит.



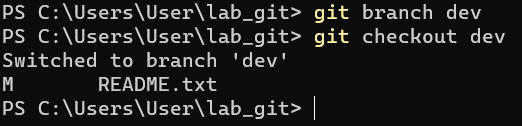
Сделаем ещё один коммит, чтобы показать ещё один способ отката.



Воспользуемся командой "git reset" она откатывает коммиты не сохраняя их.



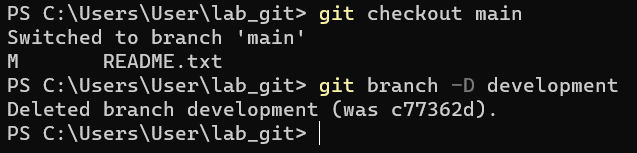
Теперь научимся работать с ветками в git. Создадим новую ветку "dev". Команда: "git branch".



Теперь с помощью команды "git branch -m" переименуем созданную ветку.

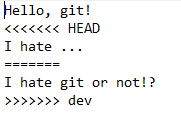


Теперь удалим эту ветку, командой "git branch -D". Для этого необходимо перейти на main ветку.



Следующим шагом опять создадим ветку и сделаем в ней коммит проведя следующие изменения. В main ветке тоже изменим эту строчку и сделаем коммит. Попробуем объединить ветки командой "git merge", возник конфликт.

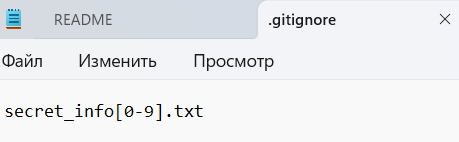


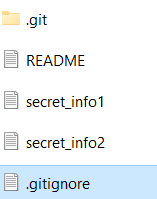


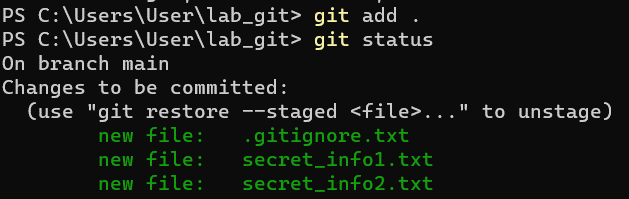
Разрешим конфликт. В файле нужно выбрать то, что мы хотим оставить в итоге: строку из dev или из master ветки. Теперь нужно просто сделать коммит в master и две ветки будут объединены.



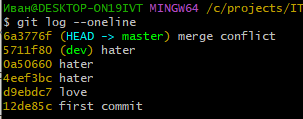
Далее познакомимся с функционалом «.gitignore». Создадим два файла secret\_data1 и secret\_data2, которые мы не хотели бы отслеживать для этого в создаём и добавляем в файл «.gitignore» добавим следующую строку.



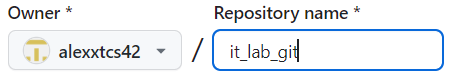




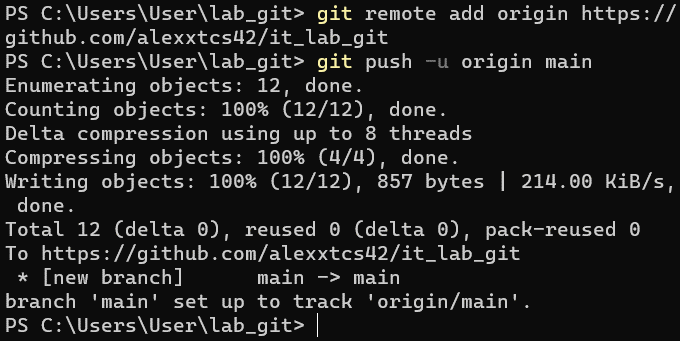
Теперь попробуем вывести лог коммитов в пользовательского формате. Команда: "git log --...".



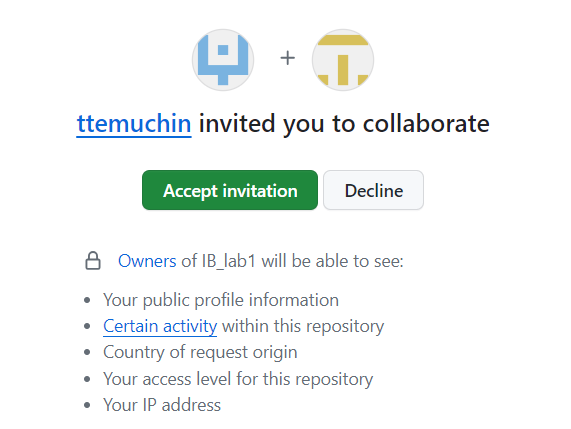
Теперь если мы хотим работать в этом репозитории с командой или хотим улучшить безопасность, необходимо создать удалённый репозиторий.



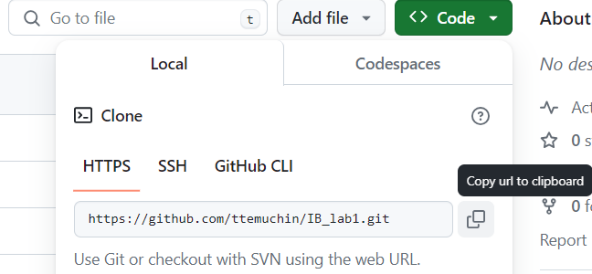
Привязываем локальный репозиторий к удалённому. Для этого выполним команду “git remote add”. Затем пушим наш проект в удалённый репозиторий.

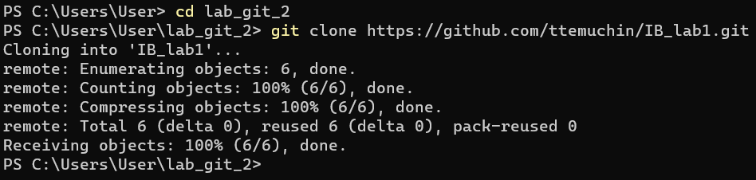


Теперь научимся совместно работать в одном репозитории. Для этого попросим коллегу пригласить нас в репозиторий.

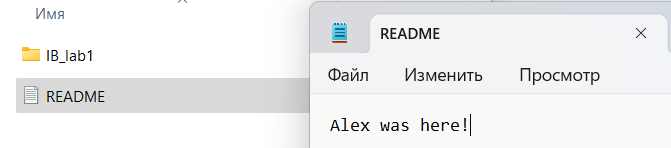


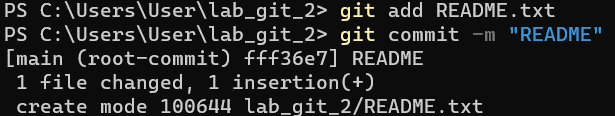
Склонируем удалённый репозиторий.

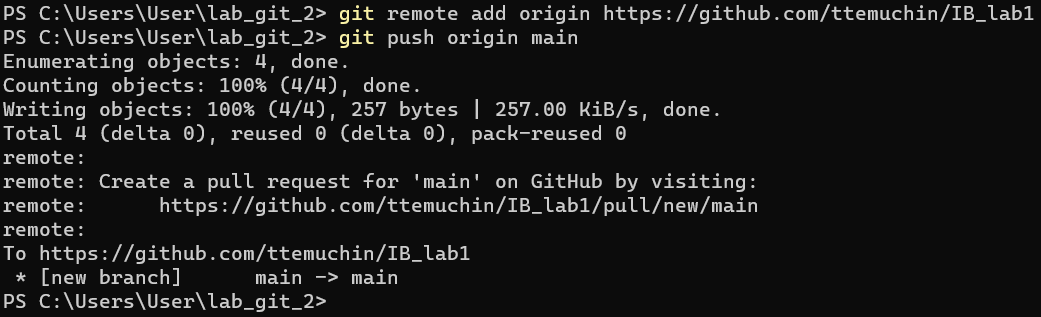




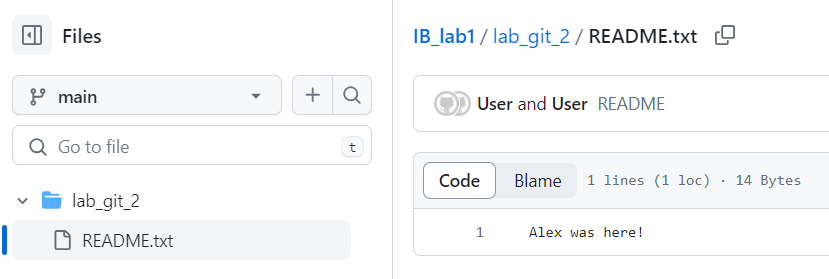
Добавляем изменения, коммитим и пушим изменения в удалённый репозиторий.







Изменения внеслись успешно.



**Вывод:** мы изучили основные возможности и получили навыки использования распределённой системы управления версиями Git, а также научились совместной работе в репозитории.