**Основы Git**

Цель – изучить и закрепить базовые навыки работы с системой контроля версий Git, включая создание локального и удалённого репозитория, управление версиями файлов, работу с ветками и слиянием изменений, а также освоить публикацию кода в GitHub.git

Задание:

Создать Git репозиторий, протестировать из практики все функции, и отправить его в удалённый репозиторий. В репозиторий отправить отчёт по выполнении данной практики.

Ход работы:

Для начала осуществляется проверка, установлен ли сам Git. В ходе выполнения команды отображается установленная версия. Все это представлено на рисунке 1.

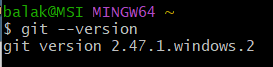


Рисунок 1 – Результат выполнения git --version

Далее нам нужно настроить опции пользователя, такие, как имя и почта. Результат выполнения соответствующих команд представлен на рисунке 2.

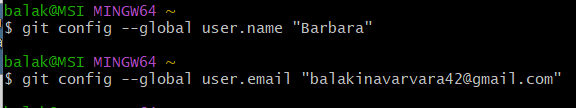


Рисунок 2 – Результат выполнения настройки пользователя

Для просмотра измененных настроек системы используется команда git config --list. Результат выполнения представлен на рисунке 3.

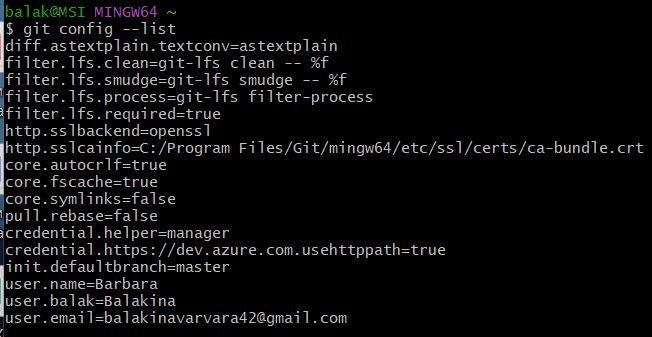


Рисунок 3 – Результат выполнения git config --list

Далее создается папка git\_practica и инициализируется репозиторий. Результат выполнения команд приведен на рисунке 4.

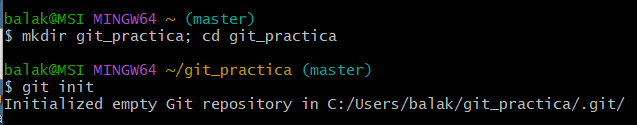


Рисунок 4 – Результат выполнения mkdir git\_practica; cd git\_practica

Далее прописывается git status, чтобы определить состояние репозитория. Все это изображено на рисунке 5.

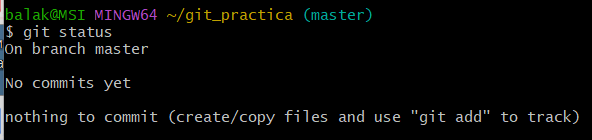


Рисунок 5 – Результат выполнения git status

Далее создадим текстовый файл с описанием внутри. Это показано на рисунке 6.



Рисунок 6 – Результат создания txt файла

Потом снова производится проверка статуса и теперь видно, что в репозитории есть текстовый файл. Это изображено на рисунке 7.

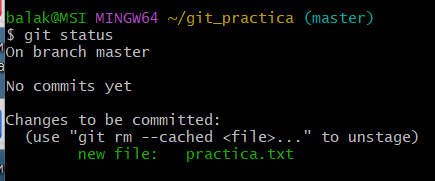


Рисунок 7 – Проверка статуса репозитория

Далее добавим коммит в репозитории. Чтобы это сделать пропишем команду git commit -m «текст коммита». Все это показано на рисунке 8.

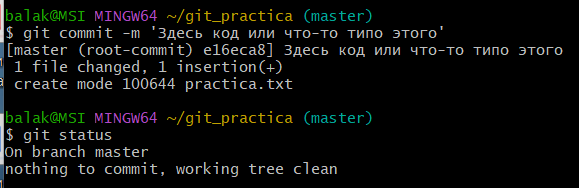


Рисунок 8 – Результат добавления однострочного сообщения

Далее осуществляется просмотр всех выполненных фиксаций с помощью команды git log. Таким образом, при выполнении программы видно, что добавлен коммит. Это показано на рисунке 9.

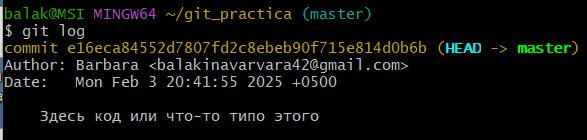


Рисунок 9 – Результат просмотра всех выполненных фиксаций

Далее с помощью команды git commit --amend -m ‘Новый комментарий’ происходит внесение нового комментария. Потом так же с помощью команды git log осуществляется просмотр изменений. Это приведено на рисунке 10.

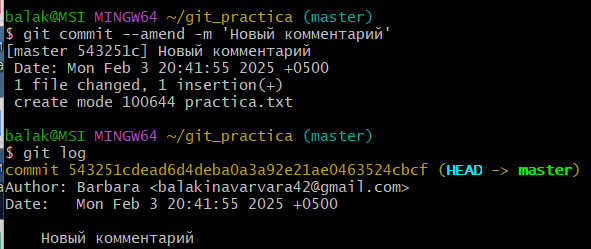


Рисунок 10 – Результат выполнения команды git commit --amend -m ‘Новый комментарий’

Далее происходит клонирование заранее созданного репозитория на Github. Создается подключение к удаленному репозиторию remote. Так же с помощью команды push происходит отправка всех изменений в удаленный репозиторий. Это представлено на рисунке 11.

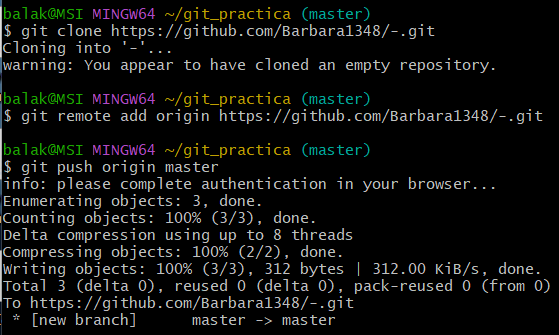


Рисунок 11 – Клонирование, подключение и отправка изменений

С помощью команды pull осуществляется запрос на изменения с сервера. Эта команда представлена на рисунке 12.

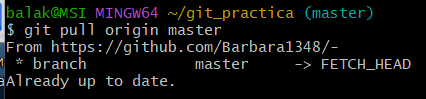


Рисунок 12 – Запрос на изменения

Для того, чтобы создать новую ветку в Git, пропишем git branch first\_new\_feature. Результат приведен на рисунке 13.



Рисунок 13 – Создание новой ветки

Также происходит переключение на эту ветку с помощью git checkout first\_new\_feature. Это показано на рисунке 14.

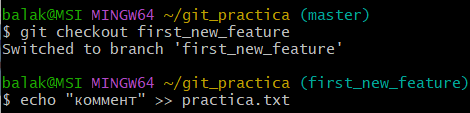


Рисунок 14 – Переключение на ветку

Теперь закоммитим эту ветку, а потом переключимся на основную ветку. Так же сделаем ветку master актуальной. Эти действия приведены на рисунке 15.

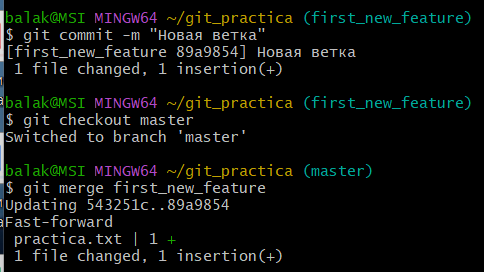


Рисунок 15 – Ветка master актуальна

Теперь происходит удаление ветки first\_new\_feature. Это показано на рисунке 16.

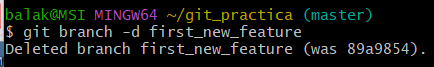


Рисунок 16 – Удаление ветки first\_new\_feature

Далее с помощью команды git log осуществляется просмотр изменений. Это приведено на рисунке 17.

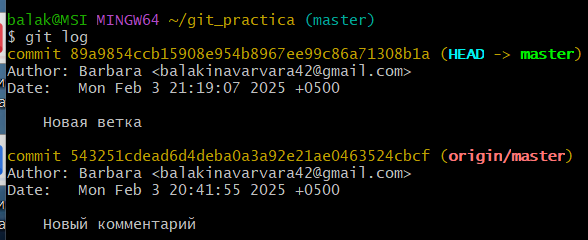


Рисунок 17 – Просмотр изменений

Теперь с помощью команды show просмотрим изменения в коммите. Это показано на рисунке 18.

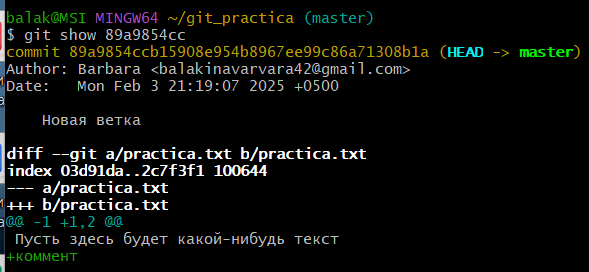


Рисунок 18 – Просмотр изменений в коммите

Чтобы увидеть разницу между двумя коммитами, используем команду diff. Это показано на рисунке 19.

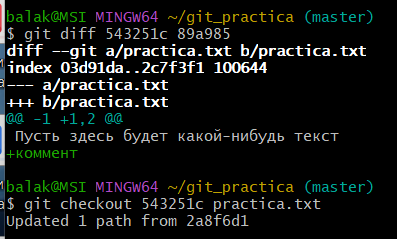


Рисунок 19 – Просмотр изменений между двумя коммитами

Далее используется команда revert, которая создает коммит, отменяющий изменения, совершенные в коммит с заданным идентификатором. Это показано на рисунке 20.

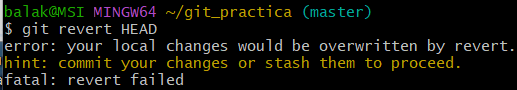


Рисунок 20 – Отмена изменений

Так же удаленный репозиторий представлен на рисунке 21.

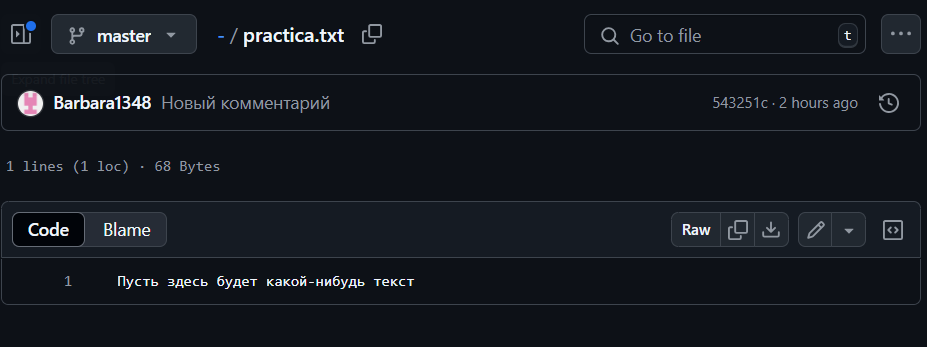


Рисунок 21 – Удаленный репозиторий