Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Отчет по лабораторной работе №5**

Дисциплина: «Разработка профессиональных приложений»

GIT

Вариант №11

Выполнил:

студент группы ИВТАСбд-22

Курдюк В.С.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

Задание по варианту

Все дальнейшие действия необходимо сделать для исходных кодов, которые были получены на этапе выполнения лабораторной работы №4.

Задание:

1. Установить Git (https://git-scm.com/)

2. Командами в КОНСОЛИ:

• 2.1. Создать локальный репозиторий

• 2.2. Добавить несколько файлов

• 2.3. Сделать один из файлов игнорируемым

• 2.4. Зафиксировать изменения

• 2.5. Внести изменения в файлы и зафиксировать изменения

• 2.6. Создать несколько бранчей, редактировать файлы в разных бранчах и фиксировать изменения. Осуществить слияние двух веток в одну.

• 2.7. Создать аккаунт github (если его нет, а если уже имеется, то использовать его), отправить исходные коды в удаленный репозиторий.

• 2.8. Склонировать удаленный репозиторий в папку, отличную от оригинальной.

• 2.9. Добиться конфликта файлов при применении изменений в оригинальном репозитории.

• 2.10. Разрешить конфликт.

• 2.11 Подготовить отчет о проделанной работе и прикрепить его в качестве ответа на это задание в learn. Отчет должен содержать список выполняемых в процессе выполнения лабораторной работы команд с их описанием, а также результат выполнения команд, скриншоты страниц с удаленного репозитория, чтоб были видны ветки и история операций (адрес репозитория также необходимо предоставить).

**Описание реализации**

До выполнения данной лабораторной работы был уже создан и склонирован репозиторий по данному предмету.

В начале была скопирована папка 4 лабораторной работы в созданную папку 5 лабораторной работы, после чего использовались команды для добавления изменений в репозиторий, в котором ещё нет других веток.

Листинг 1. Использованные команды

|  |
| --- |
| cp -r 'lab4' 'lab5'  git add lab5  git commit -m init\_lab5  git push |

После чего были добавлены папки test1 и test2 с .txt файлами с соответствующими названиями папок в папке ‘5 лаба’. И также закоммичена и запушена в main.

Листинг 2. Создание тестовых папок с фалами

|  |
| --- |
| cd lab5  mkdir test1  mkdir test2  cd test1  touch test1.txt  cd ..  cd test2  touch test2.txt  cd ..  cd ..  git add lab5  git commit -m commit\_tests\_files  git push |

Далее была создана ветка test1, и добавлена в ней папка test3 также с содержимым в папке ‘5 лаба’. А также закоммичена и запушена в эту же ветку на репозиторий.

Листинг 3. Создание ветки test1 и работа в ней

|  |
| --- |
| git branch test1  git checkout test1  cd lab5  mkdir test3  cd test3  touch test3.txt  cd ..  cd ..  git add lab5  git commit -m branch\_commit1  git push origin test1 |

Следующее что было сделано, это создание ещё одной ветки test2 с добавлением файла test4.txt в папку ‘5 лаба’

Листинг 4. Создание ветки test2 и работа в ней

|  |
| --- |
| git branch test2  git checkout test2  cd lab5  touch test4.txt  cd ..  git add lab5  git commit -m branch\_commit2  git push origin test2 |

Далее было сделано слияние ветки test1 с веткой test2

Листинг 5. Слияние веток

|  |
| --- |
| git checkout test1  git merge test2 |

Далее был добавлен файл ignored.txt с содержимым в корневую папку и .gitignore, в котором был прописан путь к данному файлу, чтобы при добавлении, коммите и пуше в репозиторий он не добавлялся.

Листинг 5. Cоздание игнорируемого файла

|  |
| --- |
| cd lab5  touch ignored.txt  touch .gitignore  vi .gitignore  cd ..  git checkout main  git add lab5  git commit -m added\_ignore  git push origin main |

Конфликт файлов можно осуществить при изменении одних и тех же файлов в разных ветках, или добавления одних и тех же файлов в разных ветках.

Поскольку уже были созданы текстовые файлы в репозитории и запушены, то я в ветке test1 и test2 изменил содержание test4.txt в папке ‘5 лаба’. Каждую из них закоммитил и запушил, а потом попытался слить.

Листинг 6. Конфликт

|  |
| --- |
| git checkout test2  cd lab5  nano test4.txt  cd ..  git add lab5  git commit -m test\_conflict\_test2  git push origin test2  git checkout test1  cd lab5  nano test4.txt  cd ..  git add lab5  git commit -m test\_conflict\_test1  git push origin test1  git merge test2 |

В итоге мне вывело ошибку в терминале:

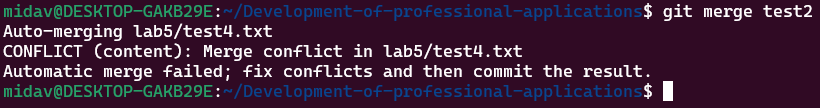


Рисунок 1. Конфликт при слиянии

Для того чтобы решить этот конфликт, нужно решить какую версию файла нужно сохранить. После чего удалить изменение, которое нам не нужно. И только потом сделать merge –continue.

в файлах появляются строки:

<<<<<<HEAD(исходный текст)

какой то текст

==========(текст приходящего слияния)

какой то текст

>>>>>>(откуда слияние)

В моём случае я удалил приходящее слияние.

**Описание возникших затруднений**

Затруднения возникли с изучением git merge и веток. И как правильно делать слияние чтобы не сломать репозиторий. Было потрачено время на изучение того как происходит выбор версий при слиянии, а также переключение между ветками и их создание.

Описание альтернативных способов решения

Вообще можно было создавать не папки с файлами для тестирования, а ещё другие проекты. А также можно было не делать простые изменения в текстовых, а делать изменения в целом программном коде.

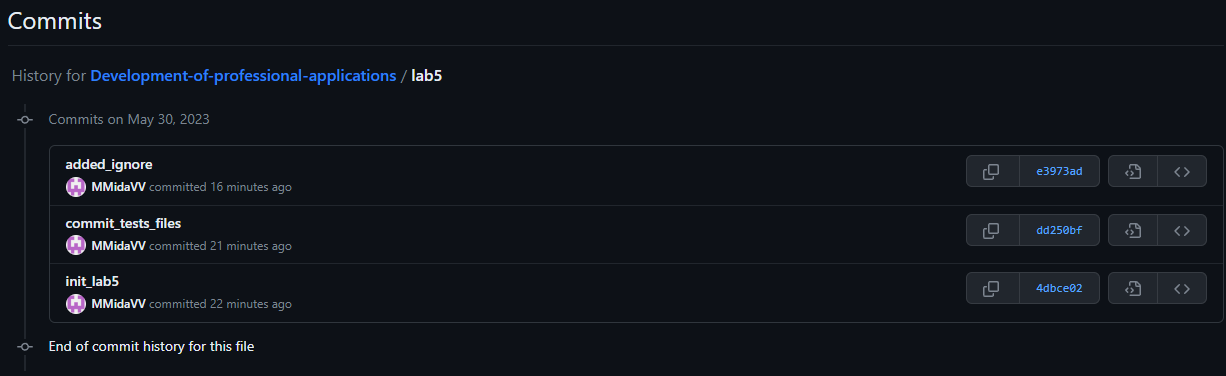
И при слиянии можно было не удалять приходящее изменения, а объединить их или оставить только приходящее изменение.

Вывод

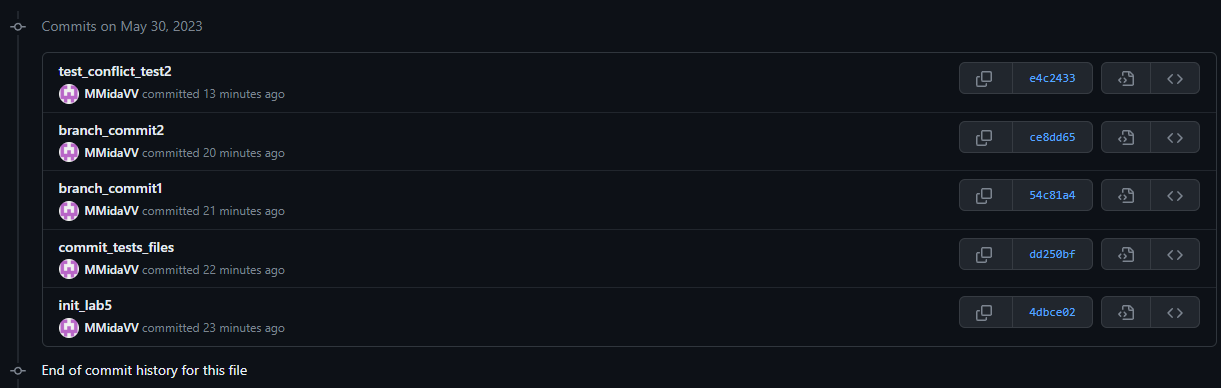
В данной лабораторной работе были приобретены навыки работы с GIT. А именно, скачивание репозитория, его загрузки, инициализации. Был изучен метод создания веток, и работа с ними. А также на практическом примере была испытана функциональность merge, с последующим решением конфликтов.

Скриншоты изменений репозитория

Ветка main:



Ветка test1:



Ветка test2:

