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 №5

Выполнил:

студент группы ИВТАСбд-21

Барыбин А.В

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 .  git commit -m init\_lab5  git push |

После чего были добавлены папки 1 и 2 с .txt файлами с соответствующими названиями папок в папке ‘lab5. И также закоммичена и запушена в main.

Далее была создана ветка 1, и добавлена в ней папка 3 также с содержимым в папке ‘lab5’. А также закоммичена и запушена в эту же ветку на репозиторий.

Листинг 2. Создание ветки test1 и работа в ней

|  |
| --- |
| git branch 1  git checkout 1  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  добавлена папка test3 с содержимым  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  git add .  git commit -m branch\_commit1  git push origin 1 |

Следующее что было сделано, это создание ещё одной ветки 2 с добавлением файла 4.txt в папку ‘lab5’

Листинг 3. Создание ветки test2 и работа в ней

|  |
| --- |
| git branch 2  git checkout 2  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  добавлен файл 4 с содержимым  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  git add .  git commit -m branch\_commit2  git push origin 2 |

Далее было сделано слияние ветки 1 с веткой 2

Листинг 4. Слияние веток

|  |
| --- |
| git checkout 1  git merge 2 |

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

|  |
| --- |
| ed ignored.txt  touch .gitignore  vi .gitignore  git checkout main  git add .  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 1 и 2 изменил содержание 4.txt в папке ‘lab5’. Каждую из них закоммитил и запушил, а потом попытался слить.

Листинг 6. Конфликт

|  |
| --- |
| git checkout 2  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  изменен файл 4  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  git add .  git commit -m test\_conflict\_2  git push origin 2  git checkout 1  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  изменен файл 4  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  git add .  git commit -m test\_conflict\_1  git push origin 1  git merge 2 |

Это вызывает ошибку

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)

….

==========(текст приходящего слияния)

….

>>>>>>(откуда слияние)

В моём случае я удалил приходящее слияние.

**Описание возникших затруднений**

Затруднения возникли с изучением git merge и веток. Пару раз был сломан 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.