Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Кафедра ИС

Отчет

По дисциплине: “Технология создания программных продуктов”

Лабораторная работа №7

“Исследование распределенных системы контроля версий Git при коллективной разработке программных продуктов”

Выполнил:

ст.гр. ИС/б-22

Долженко И.А.

Проверил:

Дрозин А. Ю.

Севастополь

2019

1 ЦЕЛЬ РАБОТЫ

Исследовать основные подходы к организации взаимодействия команды разработчиков с использованием распределенной системы контроля версий (DVCS). Приобрести практические навыки установки и настойки DVCS Git, организации ветвей разработки и осуществление слияния.

2 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Разработать модель командной работы согласно варианту, полученному у преподавателя.
2. Создать необходимое количество репозиториев, разработать соглашение по предназначению репозиториев.
3. Создать изменения в одном локальном репозитории, сохранить их в удаленном.
4. Получить набор изменений из удаленного репозитория в репозиторий отличный от описанного в п. внести дополнительные изменения и сохранить их в удаленном репозитории.
5. Внести одновременно разные изменения в локальные репозитории сохранить их все в удаленном, продемонстрировать процесс слияния.
6. Продемонстрировать создание именованных веток в локальном репозитории.
7. Проанализировать результаты работы, сделать выводы.

3 ВЫПОЛНЕНИЕ РАБОТЫ

1. Создадим локальный репозиторий.

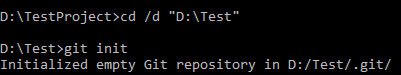


Рисунок 1 – Создание репозитория

2. Сделаем коммит текущего репозитория и выведем информацию о нем.

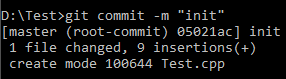


Рисунок 2 – Создание коммита

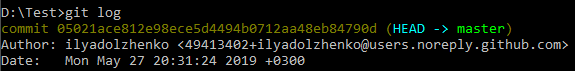


Рисунок 3 – Вывод лога

3. Свяжем локальный репозиторий с проектом и созданный удаленно на сайте BitBucket.

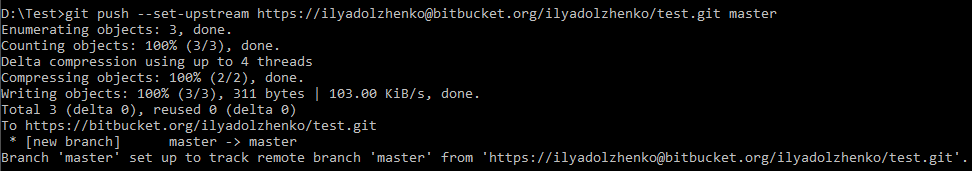


Рисунок 4 – Связывание локального и удаленного репозитория

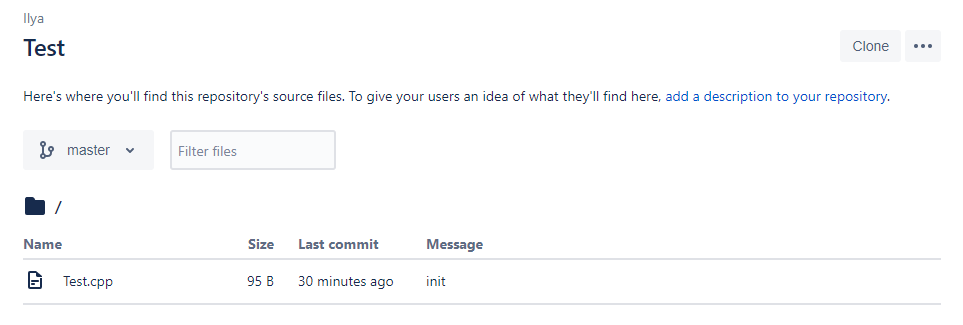


Рисунок 5 – Страница репозитория на BitBucket

4. Создадим две ветки.



Рисунок 6 – Создание веток

5. Перейдем в ветку feature и изменим в ней код.



Рисунок 7 – Переход к ветке feature

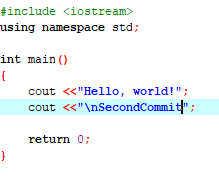


Рисунок 8 – Меняем код

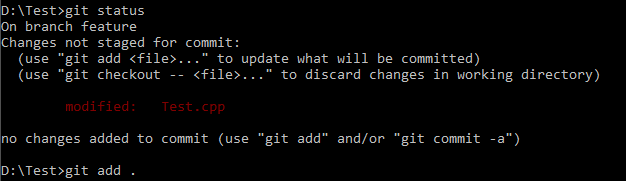


Рисунок 9 – Вывод статуса ветки feature

6. Сделаем коммит и выведем на экран лог коммитов.

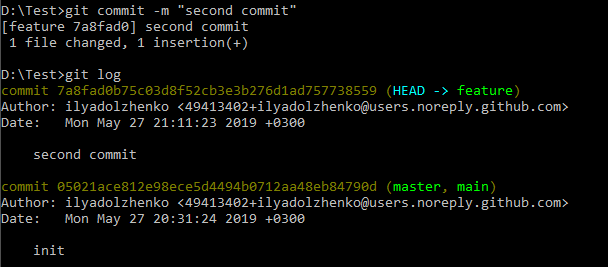


Рисунок 10 – Вывод лога

7. Перейдем в ветку main и изменим код.



Рисунок 11 – Переход в ветку main

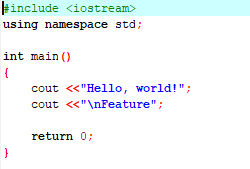


Рисунок 11 – Меняем код

8. Сделаем коммит.



Рисунок 12 – Коммит в ветке main

Для объединения веток была использована команда git merge master.

Другие команды, используемые при разработке:

- Git status - отображает текущее состояние файла;

- Git log - отображает историю коммитов;

- Git clone - создает копию основного дистрибутива;

- Git config - открывает редактор, в котором можно настроить учетную запись;

- Git add - добавляет файл в систему контроля;

- Git branches - выводит список всех локальных веток;

- Git rm - удаляет файл;

- Git mv - перемещает файл.

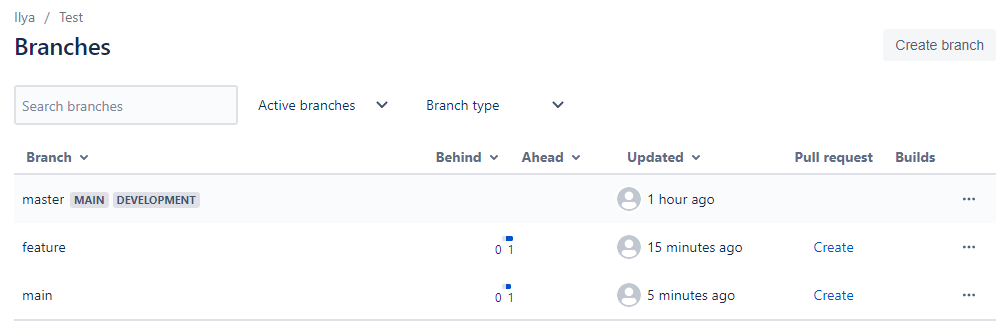


Рисунок 13 – Результат создания веток на сайте BitBucket

ВЫВОД

В ходе лабораторной работы были Исследованы основные подходы к организации взаимодействия команды разработчиков с использованием распределенной системы контроля версий (DVCS). Приобрели практические навыки установки и настройки DVCS Git, организации ветвей разработки и осуществление слияния.