**МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное**

**образовательное учреждение высшего образования**

**«СЕВЕРО - КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Языки программирования**

**Отчет по лабораторной работе №1**

**«Основы ветвления Git»**

|  |
| --- |
| Выполнил студент группы  ИТС-б-о-20-1 (1) |
| Горлов Д. С. « »\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г. |
| Подпись студента\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Работа защищена « »\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г. |
| Проверил к.т.н., доцент  Кафедры инфокоммуникаций  Воронкин Р.А.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Ставрополь 2021

**Лабораторная работа №1**

Основы ветвления Git

**Цель работы:** исследование базовых возможностей по работе с локальными и удаленными ветками Git.

**Ссылка на репозиторий:**

<https://github.com/gor-dimm/prog_lr1>

**Порядок выполнения работы:**

1. Создан общедоступный репозиторий на GitHub, в котором использована лицензия MIT.

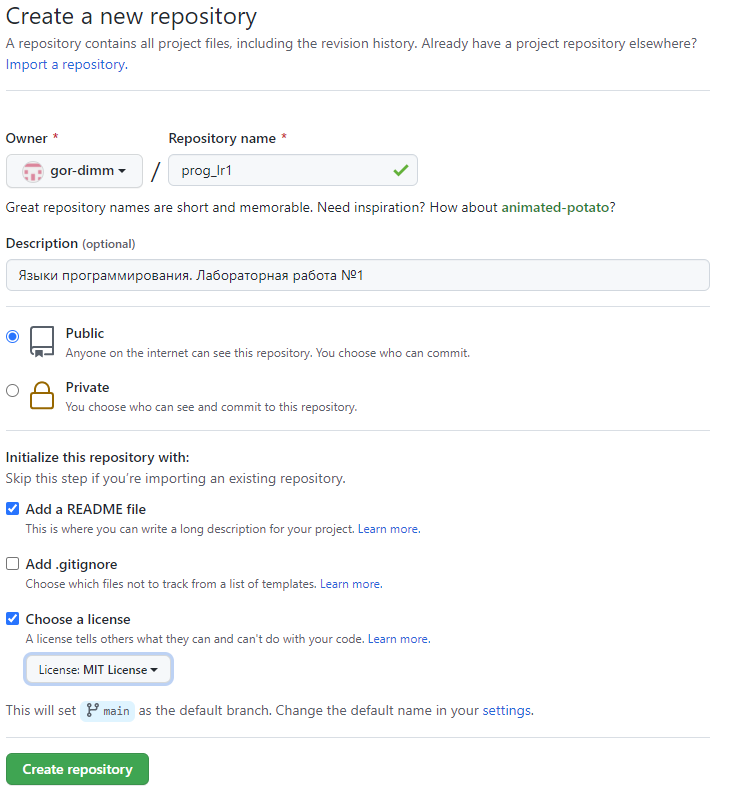


Рисунок 1. Создание общедоступного репозитория

2. В репозиторий были добавлены файлы 1.txt, 2.txt и 3.txt. Выполнена индексация файла 1.txt и коммит с комментарием «add 1.txt file».

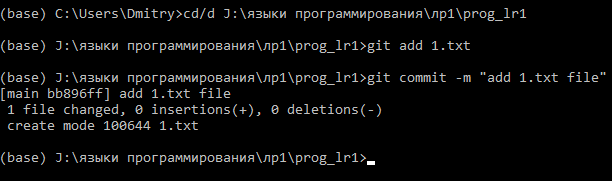


Рисунок 2. Индексация и коммит файла 1.txt

3. Выполнена индексация файлов 2.txt и 3.txt, коммит с комментарием «add 2.txt and 3.txt».

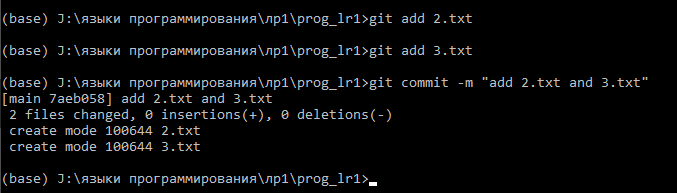


Рисунок 3. Индексация и коммит файлов 2.txt и 3.txt

4. Создана ветка «my\_first\_branch».

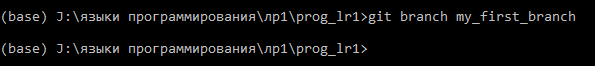


Рисунок 4. Создание новой ветки

5. Совершён переход на ветку «my first branch», создан файл in\_branch.txt, выполнен коммит.

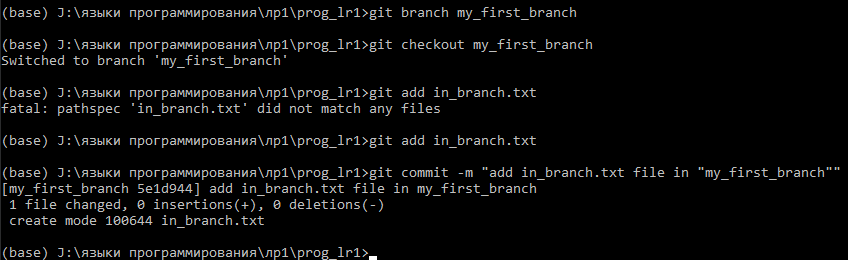


Рисунок 5. Создание файла in\_branch.txt

6. Выполнен возврат на ветку «master», создан и одновременно совершён переход на новую ветку «new\_branch».

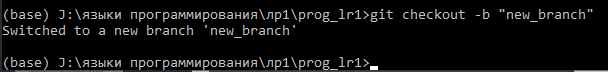


Рисунок 6. Создание и переход на ветку «new\_branch»

7. Сделаны изменения в файле 1.txt, выполнен коммит.

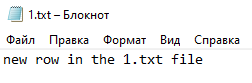


Рисунок 7. Изменение файла 1.txt

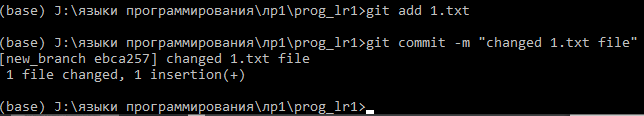


Рисунок 8. Коммит измененного файла 1.txt

8. Выполнено слияние веток master и my\_first\_branch, затем master и new\_branch.

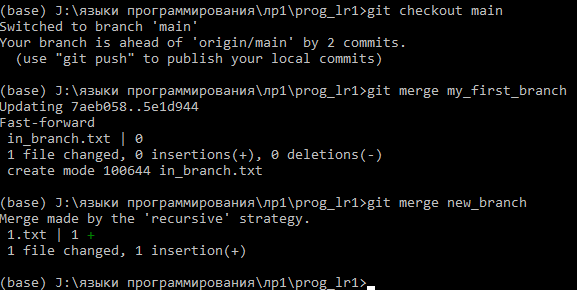


Рисунок 9. Слияние существующих веток

9. Удалены ветки «my\_first\_branch» и «new\_branch».

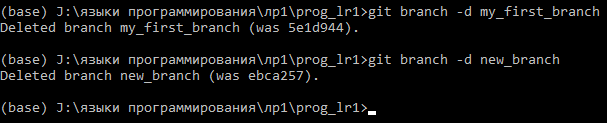


Рисунок 10. Удаление веток

10. Созданы ветки «branch\_1» и «branch\_2».

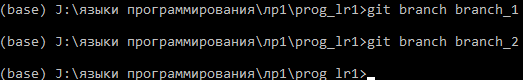


Рисунок 11. Создание веток «branch\_1» и «branch\_2»

11. Выполнен переход на ветку «branch\_1», совершены изменения в файлах 1.txt и 3.txt, выполнен коммит.

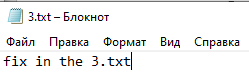
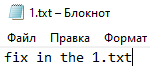


Рисунок 12. Изменение файлов 1.txt и 3.txt

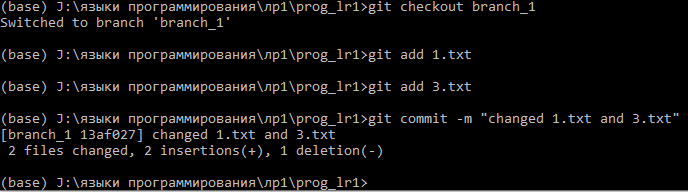


Рисунок 13. Коммит изменений

12. Совершён переход на ветку «branch\_2», изменены файлы 1.txt и 3.txt, выполнен коммит изменений.

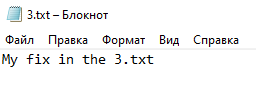
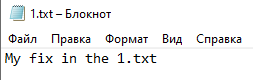


Рисунок 14. Изменение файлов 1.txt и 3.txt (My fix)

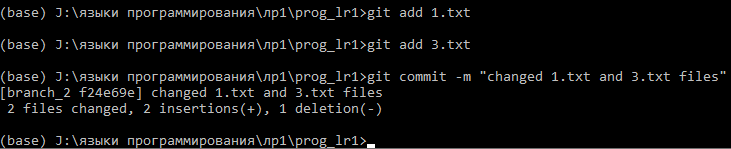


Рисунок 15. Коммит изменений в ветке 2.txt

13. Исправлены конфликты файлов, выполнен коммит и слияние веток.

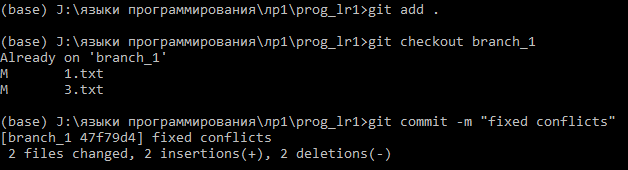




Рисунок 16. Исправление конфликта и слияние веток

14. Ветка «branch\_1» была отправлена на GitHub.

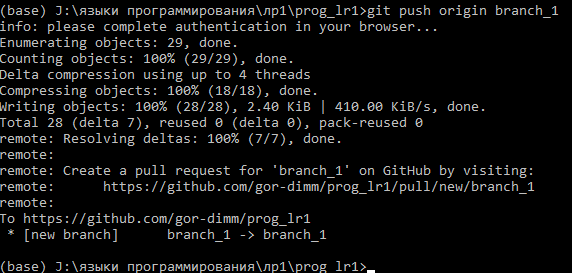


Рисунок 17. Отправка ветки на GitHub

15. Средствами GitHub была создана удалённая ветка «branch\_3».

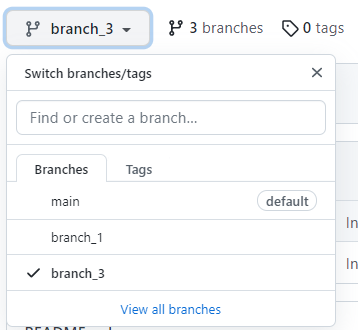


Рисунок 18. Создание удалённой ветки средствами GitHub

16. В локальном репозитории создана ветка отслеживания удалённой ветки branch\_3.

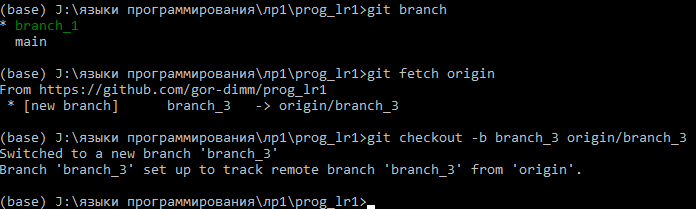


Рисунок 19. Создание ветки ослеживания

17. Выполнен переход на ветку branch\_3, добавлена строка в файл 2.txt, выполнено перемещение ветки master на ветку branch\_2. Изменения отправлены на GitHub.

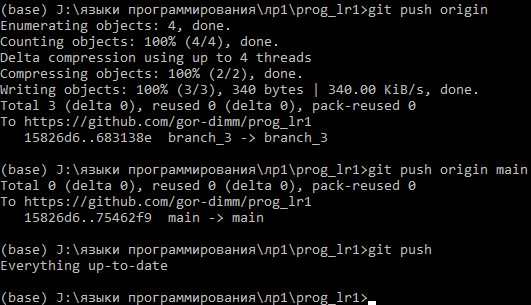


Рисунок 20. Коммит изменений и отправление веток на GitHub

**Контрольные вопросы:**

1.Что такое ветка?

Ветка - простой перемещаемый указатель на один из нескольких родительских коммитов.

2. Что такое HEAD?

HEAD - это указатель на коммит в репозитории, который станет родителем следующего коммита.

3. Способы создания веток.

Командами git branch или git checkout.

4. Как узнать текущую ветку?

Командой git log или командой git branch без указания каких-либо дополнительных параметров.

5. Как переключаться между ветками?

Командой git checkout.

6. Что такое удаленная ветка?

Удалённые ссылки — это ссылки (указатели) в удалённых репозиториях, включая ветки, теги и так далее.

7. Что такое ветка отслеживания?

Ветки слежения — это ссылки на определённое состояние удалённых веток. Это локальные ветки, которые нельзя перемещать; Git перемещает их автоматически при любой коммуникации с удаленным репозиторием, чтобы гарантировать точное соответствие с ним.

8. Как создать ветку отслеживания?

Ветка отслеживания создаётся при помощи команды git checkout –b имя\_ветки origin/имя\_ветки.

9. Как отправить изменения из локальной ветки в удаленную ветку?

Командой git push имя\_ветки.

10. В чем отличие команд git fetch и git pull?

Команда git fetch только получает данные с удалённого сервера, когда же команда git pull получает данные и выполняет слияние с веткой на рабочем месте.

11. Как удалить локальную и удаленную ветки?

Используя параметр --delete для команды git push.

12. Изучить модель ветвления git-flow (использовать материалы статей https://www.atlassian.com/ru/git/tutorials/comparing-workflows/gitflow-workflow, https://habr.com/ru/post/106912/). Какие основные типы веток присуствуют в модели git-flow? Как организована работа с ветками в модели git-flow? В чем недостатки git-flow?

Основные типы веток в модели git-flow:

- основная ветка (master);

- ветка разработки (develop);

- функциональная ветка (feature);

- ветка выпуска (release);

- ветка исправления (hotfix);

Работа с ветками в модели git-flow организована следующим образом:

1) из ветки main создается ветка develop;

2) из ветки develop создается ветка release;

3) из ветки develop создаются ветки feature;

4) когда работа над веткой feature завершается, она сливается в ветку develop;

5) когда работа над веткой release завершается, она сливается с ветками develop и main;

6) если в ветке main обнаруживается проблема, из main создается ветка hotfix;

7) когда работа над веткой hotfix завершается, она сливается с ветками develop и main.

Недостатки модели git-flow: замедление работы пользователей, сложности с частотой выпуска релизов и большие траты времени и сил в случае конфликтов.

**Вывод о проделанной работе:** проведено исследование базовых возможностей по работе с локальными и удаленными ветками Git.