МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт математики и информационных технологий имени профессора Н.И. Червякова

Кафедра инфокоммуникаций

ОТЧЕТ
по лабораторной работе №1
Дисциплина: «Языки программирования»

Выполнил студент группы

ИТС-б-о-20-1 (1)

Абдуллаев Р.Р. « »_____20__г.

Подпись студента_____

Работа защищена «

»_____20__г.

Проверил
к.т.н., доцент
кафедры инфокоммуникаций
доцент
Воронкин Р.А.

_____
(подпись)

Ставрополь, 2021 г.

ЛАБОРАТОРНАЯ РАБОТА №1

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

Ссылка на репозиторий:

https://github.com/SindiCATT/lr21.git

Порядок выполнения работы:

1. Создал три файла: 1.txt, 2.txt, 3.txt.

2. Проиндексировал и сделал коммит.

```
D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>gi
[main 96ffd22] add 1.txt file
 3 files changed, 3 insertions(+)
 create mode 100644 1.txt
 create mode 100644 2.txt
 create mode 100644 3.txt
```

Рисунок 1 – Перезапись коммита

3. Создать новую ветку my_first_branch. Создал новый файл in_branch.txt, закоммитил изменения.

```
D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git checkout -b my_first_branch
Switched to a new branch 'my_first_branch'

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>
```

Рисунок 2 – Создание новой ветки

```
D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git add my_branch.txt

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git commit -m "add my_branch.txt"
[my_first_branch d96044f] add my_branch.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 my_branch.txt

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>
```

Рисунок 3 – Добавление нового файла

4. Вернулся на ветку main, создал и сразу перенес на ветку new_branch.

```
D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git checkout -b new_branch
Switched to a new branch 'new_branch'

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>
```

Рисунок 4 – Возврат на «main» и переход на новую ветку

5. Сделал изменения в файле 1.txt, добавил “new row in the 1.txt file”, закоммитил изменения.

```
D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git add 1.txt

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git commit -m "1.txt was update"
[new_branch 9cefdf1] 1.txt was update
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>
```

Рисунок 5 – Коммит изменений первого файла

6. Выполнить слияние ветки main и my_first_branch, после ветки main и new_branch.

```
D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git merge my_first_branch
Updating 96ffd22..d96044f
Fast-forward
 my_branch.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 my_branch.txt

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git merge new_branch
Merge made by the 'ort' strategy.
 1.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>
```

Рисунок 6 – Слияние веток

7. Удаление ветки my_first_branch и new_branch.

```
D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git branch -d my_first_branch
Deleted branch my_first_branch (was d96044f).

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git branch -d new_branch
Deleted branch new_branch (was 9cefdf1).

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>S
```

Рисунок 7 – Удаление веток

8. Создание ветки branch_1 и branch_2.

```
D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git branch branch_1

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git branch branch_2
```

Рисунок 8 – Создание веток

9. Изменение файлов 1.txt и 3.txt в ветке branch_1 и коммит.

```
D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git commit -m "Изменение 1 и 3.txt в branch_1"
[main f107a54] Изменение 1 и 3.txt в branch_1
 2 files changed, 2 insertions(+), 2 deletions(-)

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>
```

Рисунок 9 – Коммит изменений

10. Изменение файлов 1.txt и 3.txt в ветке branch_2 и коммит.

```
D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git checkout branch_2
Already on 'branch_2'

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git add .

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git commit -m "Изменение 1 и 3.txt в branch_2"
[branch_2 87bf4ee] Изменение 1 и 3.txt в branch_2
 2 files changed, 2 insertions(+), 2 deletions(-)

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>
```

Рисунок 10 – Коммит изменений

11. Слияние изменения ветки branch_2 в ветку branch_1.

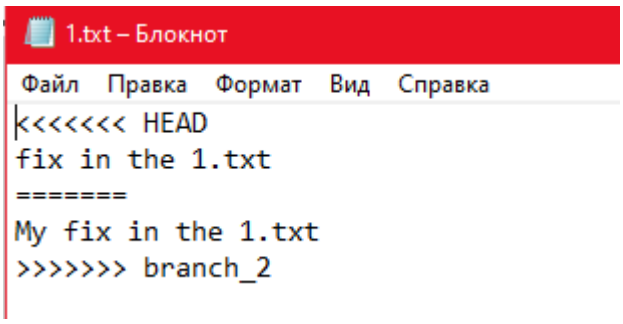
```
D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git checkout branch_1
Switched to branch 'branch_1'

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git merge branch_2
Updating 2b5bd9c..87bf4ee
Fast-forward
 1.txt | 2 +-
 3.txt | 2 +-
 2 files changed, 2 insertions(+), 2 deletions(-)

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>
```

Рисунок 11 – Слияние веток

12. Разрешение конфликта в ручном режиме.



```
D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git merge branch_2
Already up to date.
```

Рисунок 12 – Решение конфликтов

13. Отправка ветки branch_1 на GitHub. Удаление ветки branch_2.

```
D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git push origin branch_1
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 8 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (21/21), 3.45 KiB | 1.15 MiB/s, done.
Total 21 (delta 6), reused 4 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), done.
remote:
remote: Create a pull request for 'branch_1' on GitHub by visiting:
remote:      https://github.com/SindiCATT/lr21/pull/new/branch_1
remote:
To https://github.com/SindiCATT/lr21.git
 * [new branch]      branch_1 -> branch_1

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git branch -d branch_2
Deleted branch branch_2 (was 87bf4ee).
```

Рисунок 13 – Отправление на сервер и удаление ветки

14. Создание средствами GitHub удаленную ветку branch_3.

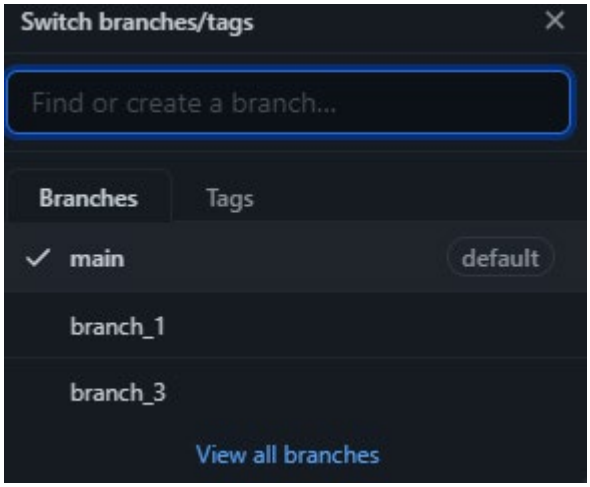


Рисунок 14 – Создание удаленной ветки

15. Добавление в файл 2.txt строку на ветке branch_3 "the final fantasy in the 4.txt file"

```
D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git add .

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>git commit -m "Добавление файла в ветку branch_3"
[branch_3 8526b8a] Добавление файла в ветку branch_3
 2 files changed, 6 insertions(+), 2 deletions(-)

D:\ДЗ\2 КУРС\языки программирования\1\lr21\lr21>
```

Рисунок 15 – Коммит файла

16. Выполнил перемещение ветки main на ветку branch_1. Отправила изменения веток main и branch_1 на GitHub.

Контрольные вопросы:

1. Что такое ветка?

Ветка в Git — это простой перемещаемый указатель на один из коммитов. По умолчанию, имя основной ветки в Git — master. Как

только вы начнёте создавать коммиты, ветка master будет всегда указывать на последний коммит. Каждый раз при создании коммита указатель ветки

master будет передвигаться на следующий коммит автоматически.

2. Что такое HEAD?

HEAD – это указатель, задача которого ссылаться на определенный коммит в репозитории.

3. Способы создания веток.

Ветки можно создать с помощью команды “git branch имя_ветки”, и чтобы перейти на неё необходимо использовать команду “git checkout имя_ветки”. Можно же создать ветку и сразу перейти на неё с помощью “git checkout –b имя_ветки”.

4. Как узнать текущую ветку?

С помощью команды git branch высветятся все ветки, текущая будет подкрашена и/или будет со знаком *.

5. Как переключаться между ветками?

Чтобы переходить по веткам необходимо использовать команду “git checkout имя_ветки”

6. Что такое удаленная ветка?

Это ветка, находящаяся на удалённом сервере GitHub.

7. Что такое ветка отслеживания?

Ветки слежения — это ссылки на определённое состояние удалённых веток. Это локальные ветки, которые напрямую связаны с удалённой веткой. Если,

находясь на ветке слежения, выполнить git pull, то Git уже будет знать с какого сервера получать данные и какую ветку использовать для слияния. При клонировании репозитория, как правило, автоматически создаётся ветка master, которая следит за origin/master.

8. Как создать ветку отслеживания?

С помощью команды git checkout -b имя_ветки origin/имя_ветки.

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

С помощью команды git push имя_ветки.

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

Git fetch лишь показывает изменения веток на сервере, но не копирует их на локальный репозиторий, в отличие от команды Git pull.

11. Как удалить локальную и удаленную ветки?

Локальную ветку можно удалить с помощью команды git branch –d имя_ветки.

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 присутствуют в модели git-flow? Как организована работа с ветками в модели git-flow? В чем недостатки git-flow?

Git Flow описывает несколько веток для разработки, релизов и взаимодействия между ними.

Репозиторий содержит 2 главные ветки:

– master;

– develop;

master — дефолтная ветка знакомая каждому. Параллельно в этой концепции существует еще одна ветка develop.

Master в этой концепции всегда содержит стабильный код, а develop существует для того чтобы от нее ответвляться и сливать туда уже готовые фичи для последующего сливания в master. Как следствие master выступает релизной веткой в этой концепции. В Git Flow мы можем использовать следующие типы веток:

– Feature branches;

– Release branches;

– Hotfix branches;

Минусы git-flow:

– git flow может замедлять работу;

– релизы сложно делать чаще, чем раз в неделю;

– большие функции могут потратить дни на конфликты и форсировать несколько циклов тестирования;

– история проекта в гите имеет кучу merge commits и затрудняет просмотр реальной работы;

– может быть проблематичным в CI/CD сценариях;

13. На прошлой лабораторной работе было задание выбрать одно из программных средств с GUI для работы с Git. Необходимо в рамках этого вопроса привести описание инструментов для работы с ветками Git, предоставляемых этим средством. Sourcetree позволяет работать с теми же инструментами, которые представление в строке CMD, но при этом управление здесь намного интуитивно понятнее и проще:

– клонирование: копирование удаленного репозитория из Bitbucket Cloud в локальную систему.

– добавление или индексирование: принятие внесенных изменений и их подготовка к добавлению в историю Git.

– коммит: добавление новых или измененных файлов в историю Git для репозитория.

– pull (извлечение): добавление в локальный репозиторий новых изменений, внесенных в репозиторий другими разработчиками.

– push: отправка изменений из локальной системы в удаленный репозиторий

Вывод: при выполнении заданий были исследованы базовые возможности работы с локальными и удаленными ветками Git.