

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙ-
СКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №1.3

Тема: «Основы ветвления git»

Выполнил студент группы

ИВТ-б-о-21-1

Богадунов В.И. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций, старший
преподаватель

Воронкин Р.А.



(подпись)

Ставрополь 2022

1. Создал репозиторий с использованием лицензии MIT. Затем клонировал его на ПК:


A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Owner * Repository name *

 ItsMyLife1337 / 

Great repository names are short, lowercase, and contain only numbers, letters, and hyphens. ForPrograms1.3 is available. Inspiration? How about **automatic-meme**?

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)


 You are creating a public repository in your personal account.

Рисунок 1.1 – Создание репозитория

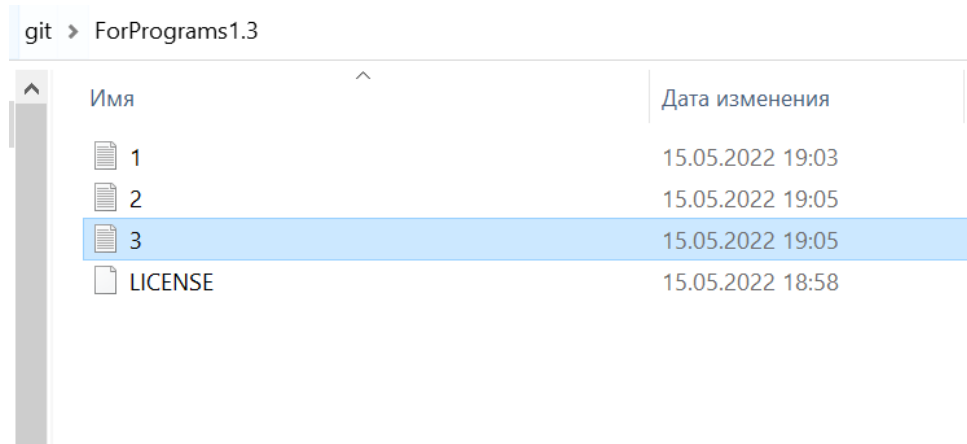
```
C:\Users\Admin>cd /d c:\users\admin\desktop\git

c:\Users\Admin\Desktop\git>git clone https://github.com/ItsMyLife1337/ForPrograms1.3.git
Cloning into 'ForPrograms1.3'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

c:\Users\Admin\Desktop\git>
```

Рисунок 1.2 – Клонирование репозитория на компьютер

2. Создал три файла, проиндексировал 1-й и сделал коммит “add 1.txt file”. Затем проиндексировал 2-й и 3-й и перезаписал коммит с новым комментарием “add 2.txt and 3.txt”:



Имя	Дата изменения
1	15.05.2022 19:03
2	15.05.2022 19:05
3	15.05.2022 19:05
LICENSE	15.05.2022 18:58

Рисунок 2.1 – Созданные файлы в репозитории

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git add 1.txt

c:\Users\Admin\Desktop\git\ForPrograms1.3>git commit -m "add 1.txt file"
[main 2062291] add 1.txt file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt

c:\Users\Admin\Desktop\git\ForPrograms1.3>
```

Рисунок 2.2 – 1-й Коммит

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git add .

c:\Users\Admin\Desktop\git\ForPrograms1.3>git commit --amend -m "add 2.txt and 3.txt"
[main dbe9b19] add 2.txt and 3.txt
Date: Sun May 15 19:05:26 2022 +0300
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt
create mode 100644 2.txt
create mode 100644 3.txt

c:\Users\Admin\Desktop\git\ForPrograms1.3>
```

Рисунок 2.3 – Изменение 1-го коммита

3. Создал новую ветку `my_first_branch`, перешел на нее и создал в ней новый файл `in_branch.txt` и сделал коммит. После этого вернулся на ветку `main`, создал и сразу перешел на ветку `new_branch`, сделал изменения в файле `1.txt` и коммит изменения.

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git checkout my_first_branch
Switched to branch 'my_first_branch'

c:\Users\Admin\Desktop\git\ForPrograms1.3>git branch
  main
* my_first_branch

c:\Users\Admin\Desktop\git\ForPrograms1.3>
```

Рисунок 3.1 – Создание и переход на новую ветку

git > ForPrograms1.3

Имя	Дата изменения
1	15.05.2022 19:03
2	15.05.2022 19:05
3	15.05.2022 19:05
in_branch	15.05.2022 19:15
LICENSE	15.05.2022 18:58

Рисунок 3.2 – Добавление файла в репозиторий

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git add .

c:\Users\Admin\Desktop\git\ForPrograms1.3>git commit -m "add in_branch.txt"
[my_first_branch 5412a6a] add in_branch.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 in_branch.txt

c:\Users\Admin\Desktop\git\ForPrograms1.3>
```

Рисунок 3.3 – Сделал коммит изменений


```
c:\Users\Admin\Desktop\git\ForPrograms1.3>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)

c:\Users\Admin\Desktop\git\ForPrograms1.3>git checkout -b new_branch
Switched to a new branch 'new_branch'

c:\Users\Admin\Desktop\git\ForPrograms1.3>git branch
  main
  my_first_branch
* new_branch

c:\Users\Admin\Desktop\git\ForPrograms1.3>
```

Рисунок 3.4 – Переход на ветку `main` и создание + переход на новую ветку

 *1 – Блокнот

Файл Правка Формат Вид Справка
new row in the 1.txt|

Рисунок 3.5 – Изменения в файле 1.txt

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git add .  
  
c:\Users\Admin\Desktop\git\ForPrograms1.3>git commit -am "fix 1.txt"  
[new_branch 1e4ab7a] fix 1.txt  
1 file changed, 1 insertion(+)
```

Рисунок 3.6 – Коммит изменений

4. Перешел на ветку main и слил ветки main и my_first_branch, после этого слил ветки main и new_branch. Удалил все ветки кроме main.

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>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)  
  
c:\Users\Admin\Desktop\git\ForPrograms1.3>git merge my_first_branch  
Updating dbe9b19..5412a6a  
Fast-forward  
  in_branch.txt | 0  
  1 file changed, 0 insertions(+), 0 deletions(-)  
  create mode 100644 in_branch.txt  
  
c:\Users\Admin\Desktop\git\ForPrograms1.3>git branch  
* main  
  my_first_branch  
  new_branch  
  
c:\Users\Admin\Desktop\git\ForPrograms1.3>git merge new_branch  
Merge made by the 'ort' strategy.  
  1.txt | 1 +  
  1 file changed, 1 insertion(+)  
  
c:\Users\Admin\Desktop\git\ForPrograms1.3>git branch -d my_first_branch  
Deleted branch my_first_branch (was 5412a6a).  
  
c:\Users\Admin\Desktop\git\ForPrograms1.3>git branch -d new_branch  
Deleted branch new_branch (was 1e4ab7a).  
  
c:\Users\Admin\Desktop\git\ForPrograms1.3>
```

Рисунок 4 – Слияние и удаление веток

5. Создал ветки branch_1 и branch_2. Перешел на ветку branch_1 и изменил файлы 1.txt, 3.txt, после чего закоммитил изменения. Перешел на ветку branch_2, изменил файлы 1.txt, 3.txt, после чего закоммитил изменения. После этого слил изменения ветки branch_2 в ветку branch_1.

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git branch
  branch_1
  branch_2
* main


c:\Users\Admin\Desktop\git\ForPrograms1.3>
```


Рисунок 5.1 – Создание веток branch_1 и branch_2

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git checkout branch_1
Switched to branch 'branch_1'

c:\Users\Admin\Desktop\git\ForPrograms1.3>
```

Рисунок 5.2 – Переход на ветку branch_1

 *1 – Блокнот

 *3 – Блокнот

Файл	Правка	Формат	Вид	Файл	Правка	Формат	Вид	С
fix		in the	1.txt	fix		in the	3.txt	

Рисунок 5.3 – Изменения в файлах 1.txt и 3.txt

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git checkout branch_1
Switched to branch 'branch_1'

c:\Users\Admin\Desktop\git\ForPrograms1.3>git add .

c:\Users\Admin\Desktop\git\ForPrograms1.3>git commit -m "fix 1.txt & 3.txt in branch_1"
[branch_1 929ef63] fix 1.txt & 3.txt in branch_1
 2 files changed, 2 insertions(+), 1 deletion(-)

c:\Users\Admin\Desktop\git\ForPrograms1.3>
```

Рисунок 5.4 – Коммит изменений в ветку branch_1

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git checkout branch_2
Switched to branch 'branch_2'

c:\Users\Admin\Desktop\git\ForPrograms1.3>
```

Рисунок 5.5 – Переход на ветку branch_2

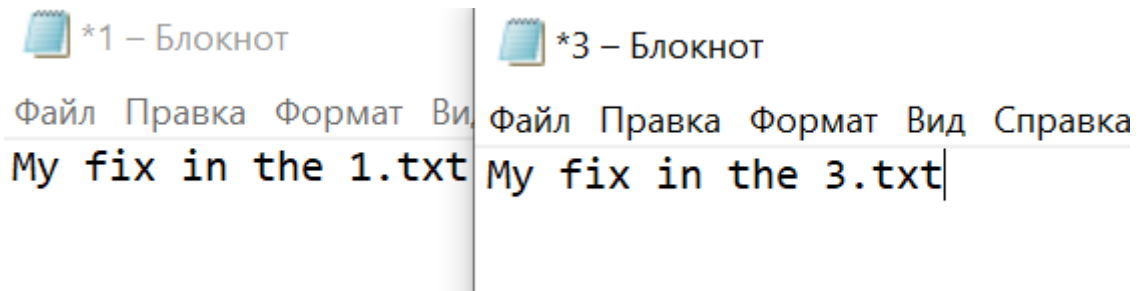


Рисунок 5.6 – Изменения в файлах 1.txt и 3.txt

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git commit -am "fix 1.txt & 3.txt in branch_2"
[branch_2 c9b631c] fix 1.txt & 3.txt in branch_2
2 files changed, 2 insertions(+), 1 deletion(-)
```

Рисунок 5.7 – Коммит изменений

6. Выполнил слияние новых веток и решил конфликт файла 1.txt в ручном режиме, а 3.txt в автоматическом инструментом vimdiff, после этого отправил ветку branch_1 в GitHub.

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git merge branch_2
Auto-merging 1.txt
CONFLICT (content): Merge conflict in 1.txt
Auto-merging 3.txt
CONFLICT (content): Merge conflict in 3.txt
Automatic merge failed; fix conflicts and then commit the result.
c:\Users\Admin\Desktop\git\ForPrograms1.3>
```

Рисунок 6.1 – Слияние веток

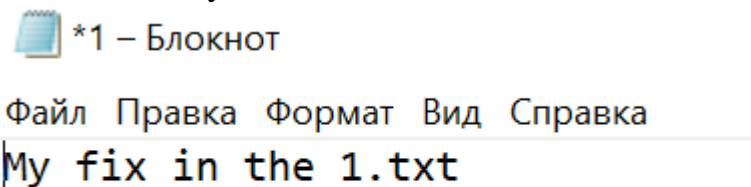


Рисунок 6.2 – Решение конфликта в ручном режиме

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git status
On branch branch_1
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Changes to be committed:
  modified:   1.txt

Unmerged paths:
  (use "git add <file>..." to mark resolution)
  both modified:   3.txt
```

Рисунок 6.3 – Добавление 1.txt в индекс, т.к. конфликт решён

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git mergetool

This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
tortoisemerge emerge vimdiff nvimdiff
Merging:
3.txt

Normal merge conflict for '3.txt':
  {local}: modified file
  {remote}: modified file
Hit return to start merge resolution tool (vimdiff):
```

Рисунок 6.4 – Команда mergetool

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git status
On branch branch_1
You have unmerged paths.
  (fix conflicts and run "git commit")
  fix in the 3.txt
<txt [unix] (20:08 15/05/2022)1,5-1 All <.txt [unix] (20:08 15/05/2022)0,0-1 All <txt [unix] (20:08 15/05/2022)1,5-1
```

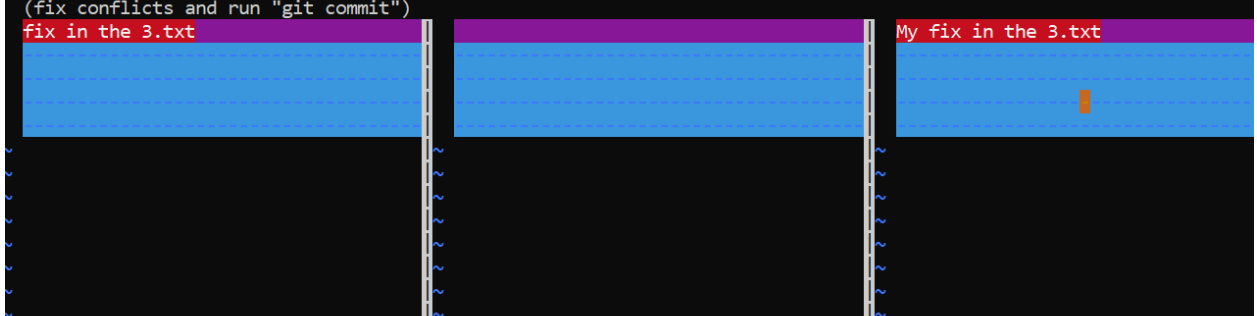


Рисунок 6.5 – Команда mergetool

```
3.txt [dos] (20:08 15/05/2022)
```

```
"3.txt" [dos] 5L, 80B
```

```
:diffg LO
```

Рисунок 6.6 – Команда diffg LO

```
[+] [dos] (20:08 15/05/2022)
```

```
"3.txt" [dos] 1L, 18B written
```

Рисунок 6.7 – Результат команды :wqa


```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git push --set-upstream origin branch_1
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 4 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (14/14), 1.16 KiB | 596.00 KiB/s, done.
Total 14 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), done.
remote:
remote: Create a pull request for 'branch_1' on GitHub by visiting:
remote:   https://github.com/ItsMyLife1337/ForPrograms1.3/pull/new/branch_1
remote:
To https://github.com/ItsMyLife1337/ForPrograms1.3.git
 * [new branch]      branch_1 -> branch_1
branch 'branch_1' set up to track 'origin/branch_1'.

c:\Users\Admin\Desktop\git\ForPrograms1.3>
```

Рисунок 6.8 – Пуш ветки на удалённый репозиторий

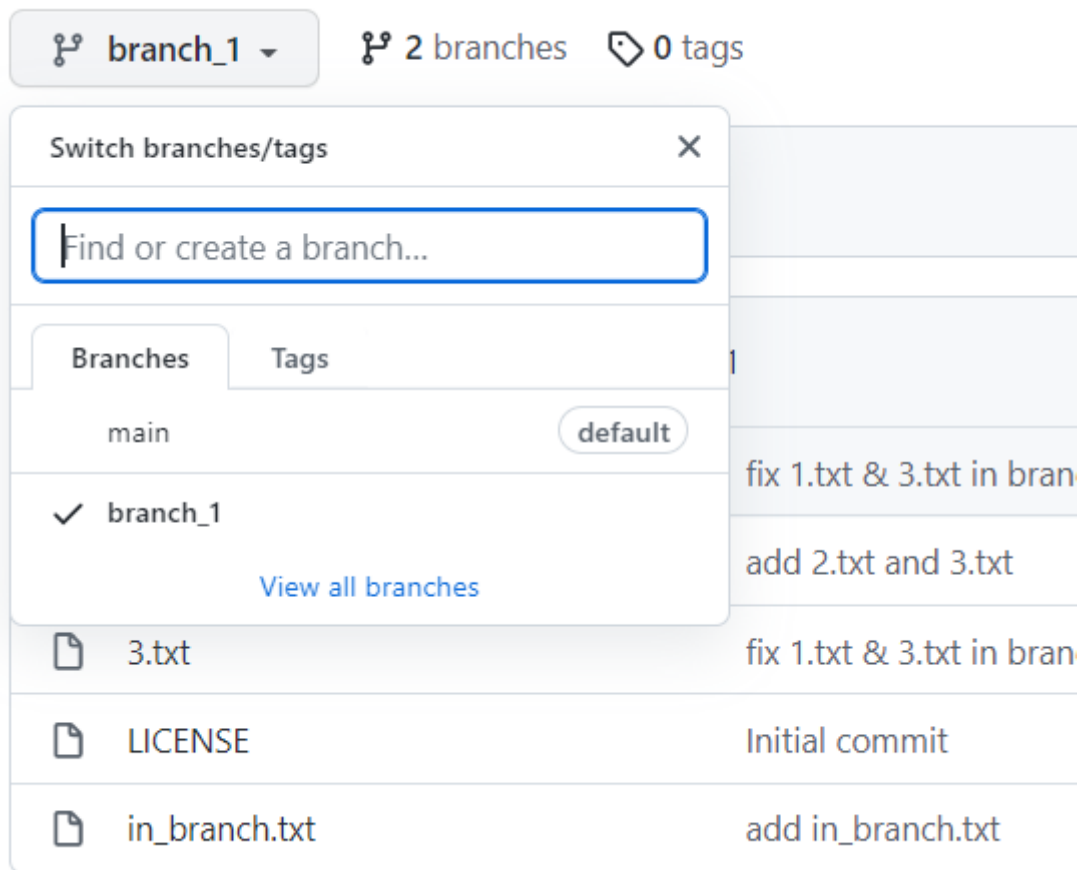


Рисунок 6.9 – Изменения на удалённом репозитории

7. Создал средствами GitHub удаленную ветку branch_3, создал в лок. Репозитории ветку отслеживания удаленной ветки branch_3. Перешел на ветку branch_3 и добавил в файл 2.txt строку “the final fantasy in the 4.txt file”.

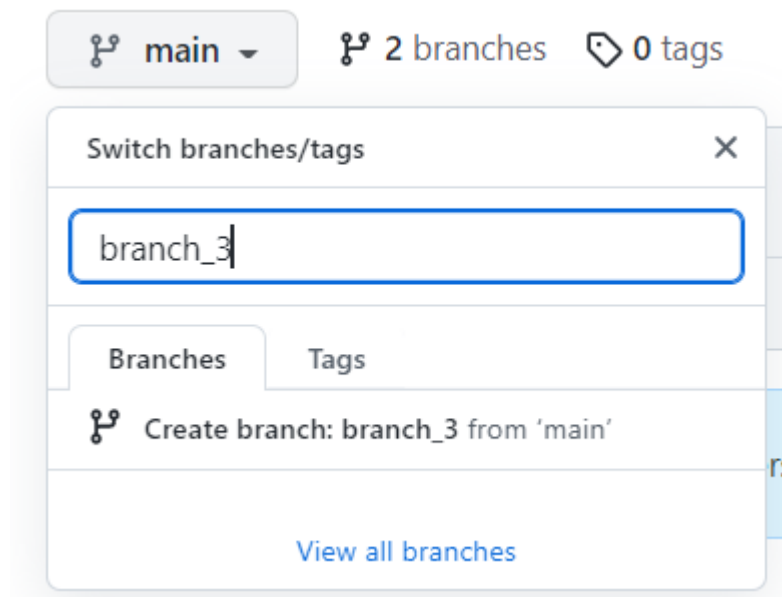


Рисунок 7.1 – Создание удалённой ветки branch_3

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git fetch --all
Fetching origin
From https://github.com/ItsMyLife1337/ForPrograms1.3
* [new branch]      branch_3    -> origin/branch_3

c:\Users\Admin\Desktop\git\ForPrograms1.3>
```

Рисунок 7.2 – Загрузка содержимого из удалённого репозитория

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git checkout --track origin/branch_3
Switched to a new branch 'branch_3'
branch 'branch_3' set up to track 'origin/branch_3'.

c:\Users\Admin\Desktop\git\ForPrograms1.3>
```

Рисунок 7.3 – Создание ветки отслеживания branch_3



2 – Блокнот

Файл Правка Формат Вид Справка

the final fantasy in the 4.txt file

Рисунок 7.4 Добавление в 2.txt строку “the final fantasy in the 4.txt file”
на ветке branch_3

8. Выполнил перемещение ветки main на ветку branch_2 и отправил изменения веток main и branch_2 на GitHub.

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git checkout branch_2
Switched to branch 'branch_2'

c:\Users\Admin\Desktop\git\ForPrograms1.3>git rebase main
Current branch branch_2 is up to date.

c:\Users\Admin\Desktop\git\ForPrograms1.3>git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 4 commits.
(use "git push" to publish your local commits)

c:\Users\Admin\Desktop\git\ForPrograms1.3>git merge branch_2
Updating 3e7e02d..c9b631c
Fast-forward
 1.txt | 2 +-
 3.txt | 1 +
 2 files changed, 2 insertions(+), 1 deletion(-)

c:\Users\Admin\Desktop\git\ForPrograms1.3>
```

Рисунок 8.1 – Результат выполнения команд

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 405 bytes | 202.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ItsMyLife1337/ForPrograms1.3.git
 2bd6e5d..c9b631c  main -> main
```

Рисунок 8.2 – Отправка изменений на удалённый репозиторий

```
c:\Users\Admin\Desktop\git\ForPrograms1.3>git push --set-upstream origin branch_2
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'branch_2' on GitHub by visiting:
remote:   https://github.com/ItsMyLife1337/ForPrograms1.3/pull/new/branch_2
remote:
To https://github.com/ItsMyLife1337/ForPrograms1.3.git
 * [new branch]      branch_2 -> branch_2
branch 'branch_2' set up to track 'origin/branch_2'.

c:\Users\Admin\Desktop\git\ForPrograms1.3>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

Рисунок 8.3 – Отправил ветку branch_2 на удалённый репозиторий

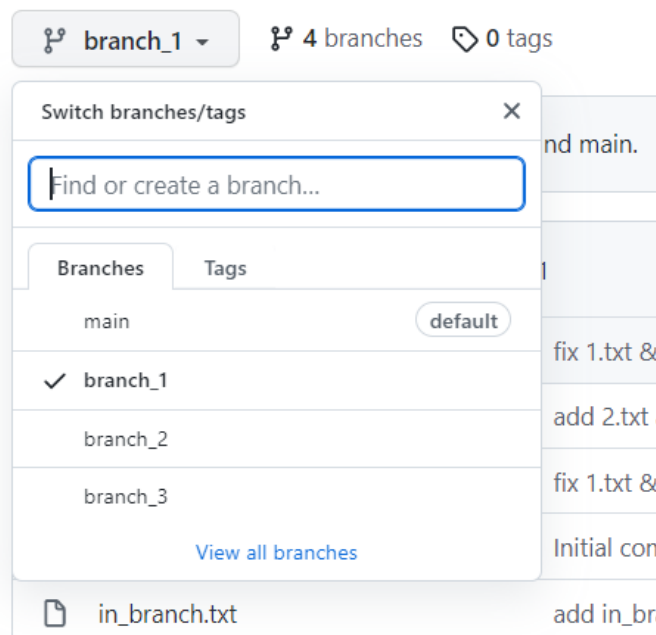


Рисунок 8.5 – Зафиксировал изменения на удалённом репозитории

Ответы на контрольные вопросы:

1) Что такое ветка?

Это последовательность коммитов, отклоняющаяся от основной линии разработки.

2) Что такое HEAD?

Это указатель, задача которого ссылаться на определённый коммит в репозитории, если точнее: на коммит, который станет родителем для следующего коммита.

3) Способы создания веток?

Новую ветку можно создать командой `git branch <название_ветки>` или на удалённом репозитории `git hub`.

4) Как узнать текущую ветку?

При помощи команды `git branch`.

5) Как переключаться между ветками?

При помощи команды `git checkout <название_ветки>`.

6) Что такое удалённая ветка?

Это ветка, находящаяся на удалённом репозитории. Или ссылка на состояние ветки на удалённом репозитории.

7) Что такое ветка отслеживания?

Это ветка в локальной репозитории, которая напрямую связана с удаленной веткой на удаленном репозитории.

8) Как создать ветку отслеживания?

Командой `git checkout --track origin/<название_ветки>`.

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

Командой `git push origin <название_ветки>`.

10) В чем отличие команд `git fetch` и `git pull`?

`Git pull` – это сочетание команд `git fetch` (получение изменений с удаленного репозитория) и `git merge` (объединение веток).

11) Как удалить локальную и удаленную ветки?

Используя команду `git branch -d <название_ветки>`. Для удаление удаленной ветки существует команда `git push origin -d <название_ветки>`.

Вывод: освоил работу с ветками в `git`.