

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Алтайский государственный технический университет им. И. И.
Ползунова»

Факультет информационных технологий
Кафедра информатики, вычислительной
техники и информационной безопасности

Отчет защищен с оценкой

Преподаватель _____ С. В. Умбетов
« _____ » _____ 2024г.

Отчёт по лабораторной работе №1 по дисциплине
«Алгоритмизация и программирование»
«Основы работы с системой контроля версий git»

ЛР 09.03.03.32.001

Студент группы ПИЭ-32
_____ *группа*

М. В. Выдрыч
_____ *и.о., фамилия*

Преподаватель ассистент, к. т. н.
_____ *должность, ученая степень*

С. В. Умбетов
_____ *и.о., фамилия*

БАРНАУЛ 2024

Цель: познакомиться с git, получить практические навыки создания, управления репозиторием.

Задание:

1. скачать git(консольная версия)
2. создать репозиторий vydrych_pie_32_a
3. создать не менее 3 коммитов в репозитории (коммит - комментирование, точка сохранения)
4. создать новую ветку в репозитории (репозиторий - хранилище)
5. создать не менее 2 коммитов в этой ветке (ветка - репозиторий в репозитории)
6. произвести слияние веток
7. вывести статус коммитов
8. скачать гитхаб десктоп
9. используя гитхаб десктоп сделать лабу еще раз, но репозиторий должен быть удаленным

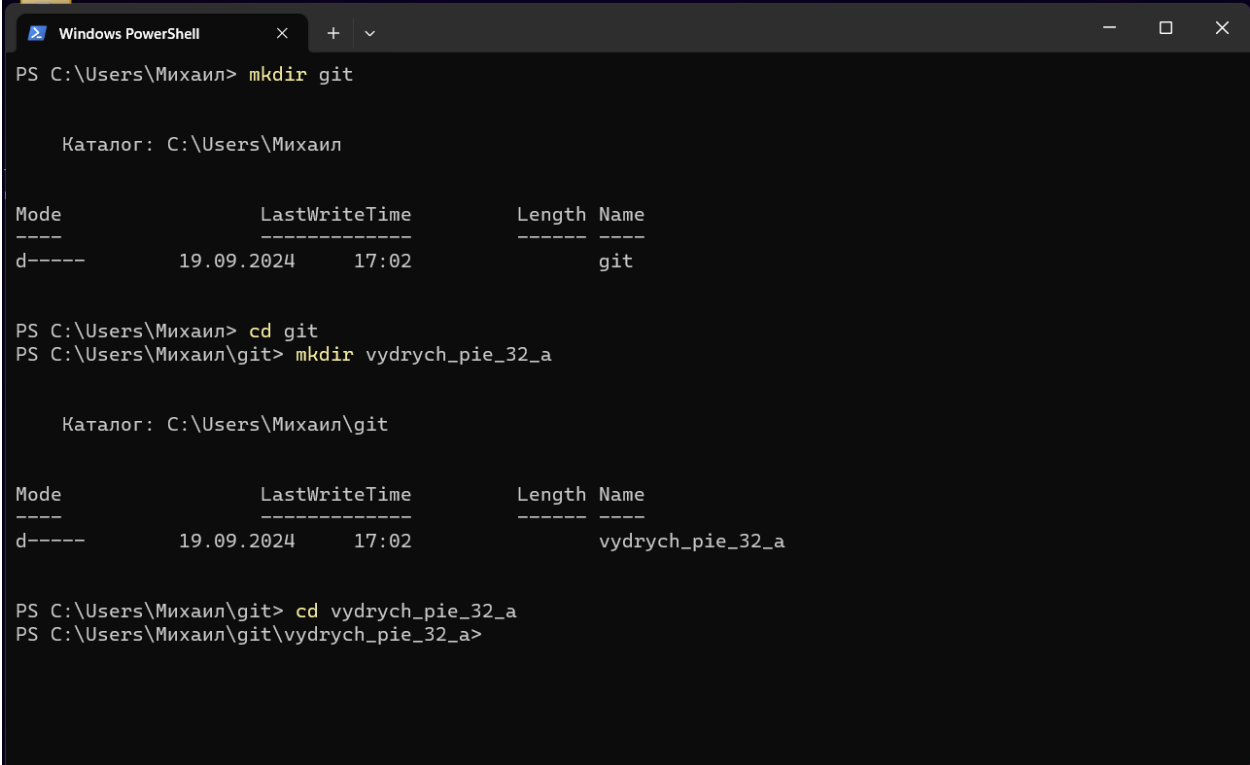


Задание принял

Выдрыч М. В

Ход работы:

Создаем директории командой `mkdir` и переходим в последнюю командой `cd`



```
Windows PowerShell
PS C:\Users\Михаил> mkdir git

Каталог: C:\Users\Михаил

Mode                LastWriteTime         Length Name
----                -
d-----          19.09.2024         17:02         git

PS C:\Users\Михаил> cd git
PS C:\Users\Михаил\git> mkdir vydrych_pie_32_a

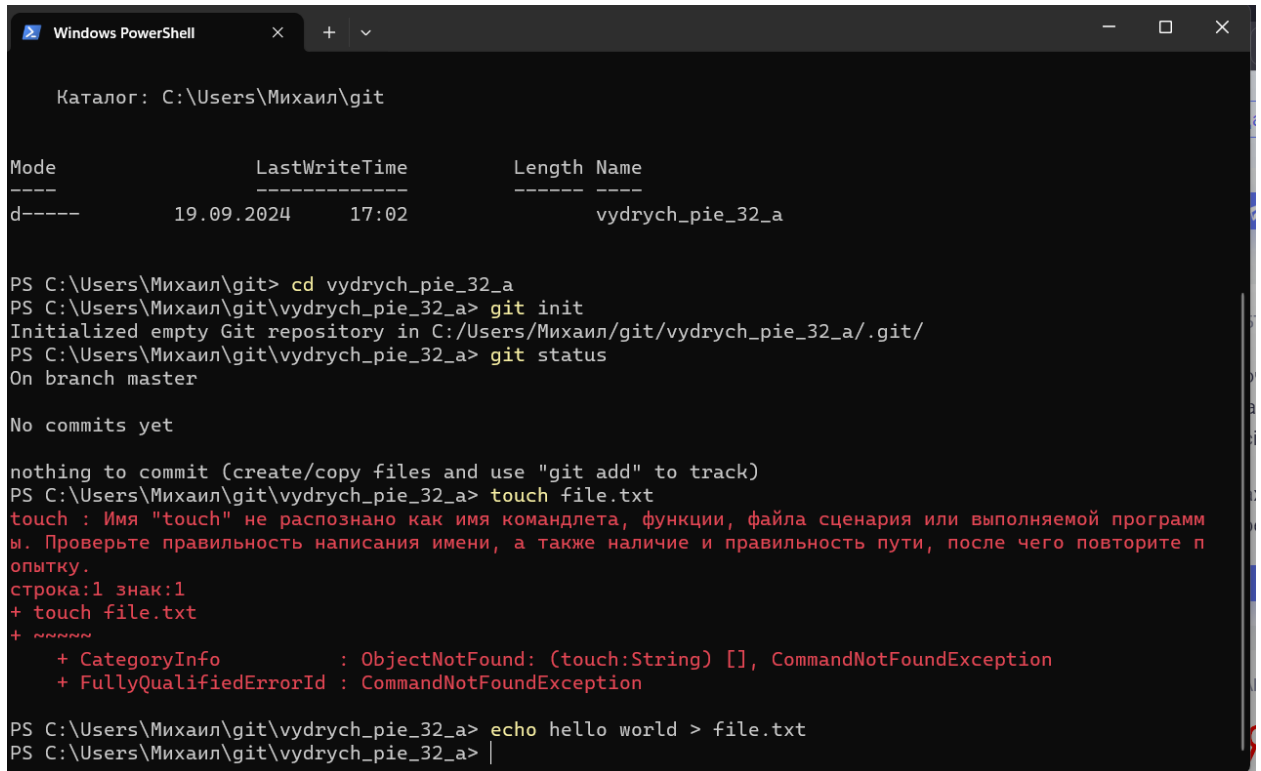
Каталог: C:\Users\Михаил\git

Mode                LastWriteTime         Length Name
----                -
d-----          19.09.2024         17:02         vydrych_pie_32_a

PS C:\Users\Михаил\git> cd vydrych_pie_32_a
PS C:\Users\Михаил\git\vydrych_pie_32_a>
```

Рисунок 1 – Создание директорий

Создаем репозиторий командой `git init` и файл командой `echo`.



```
Каталог: C:\Users\Михаил\git

Mode                LastWriteTime         Length Name
----                -
d-----          19.09.2024    17:02             vydrych_pie_32_a

PS C:\Users\Михаил\git> cd vydrych_pie_32_a
PS C:\Users\Михаил\git\vydrych_pie_32_a> git init
Initialized empty Git repository in C:/Users/Михаил/git/vydrych_pie_32_a/.git/
PS C:\Users\Михаил\git\vydrych_pie_32_a> git status
On branch master

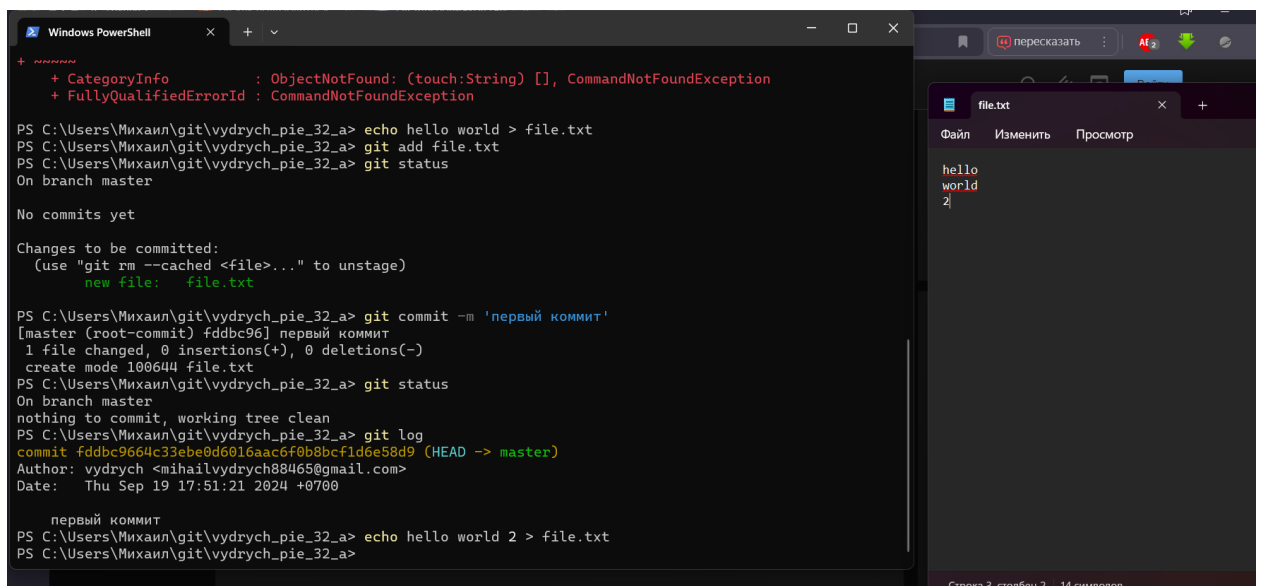
No commits yet

nothing to commit (create/copy files and use "git add" to track)
PS C:\Users\Михаил\git\vydrych_pie_32_a> touch file.txt
touch : Имя "touch" не распознано как имя командлета, функции, файла сценария или выполняемой программ
ы. Проверьте правильность написания имени, а также наличие и правильность пути, после чего повторите п
опытку.
строка:1 знак:1
+ touch file.txt
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (touch:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\Михаил\git\vydrych_pie_32_a> echo hello world > file.txt
PS C:\Users\Михаил\git\vydrych_pie_32_a> |
```

Рисунок 2 – Инициализация репозитория

Добавляем файл в коммит командой `git add` и создаем сам коммит командой `git add` и `git commit`. Следующей командой `echo` вносим изменения в файл.



```
+ CategoryInfo          : ObjectNotFound: (touch:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\Михаил\git\vydrych_pie_32_a> echo hello world > file.txt
PS C:\Users\Михаил\git\vydrych_pie_32_a> git add file.txt
PS C:\Users\Михаил\git\vydrych_pie_32_a> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file.txt

PS C:\Users\Михаил\git\vydrych_pie_32_a> git commit -m 'первый коммит'
[master (root-commit) fddbc96] первый коммит
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file.txt
PS C:\Users\Михаил\git\vydrych_pie_32_a> git status
On branch master
nothing to commit, working tree clean
PS C:\Users\Михаил\git\vydrych_pie_32_a> git log
commit fddbc9664c33ebe0d6016aac6f0b8bcfd6e58d9 (HEAD -> master)
Author: vydrych <mihailvydrych88465@gmail.com>
Date:   Thu Sep 19 17:51:21 2024 +0700

    первый коммит

PS C:\Users\Михаил\git\vydrych_pie_32_a> echo hello world 2 > file.txt
PS C:\Users\Михаил\git\vydrych_pie_32_a>
```

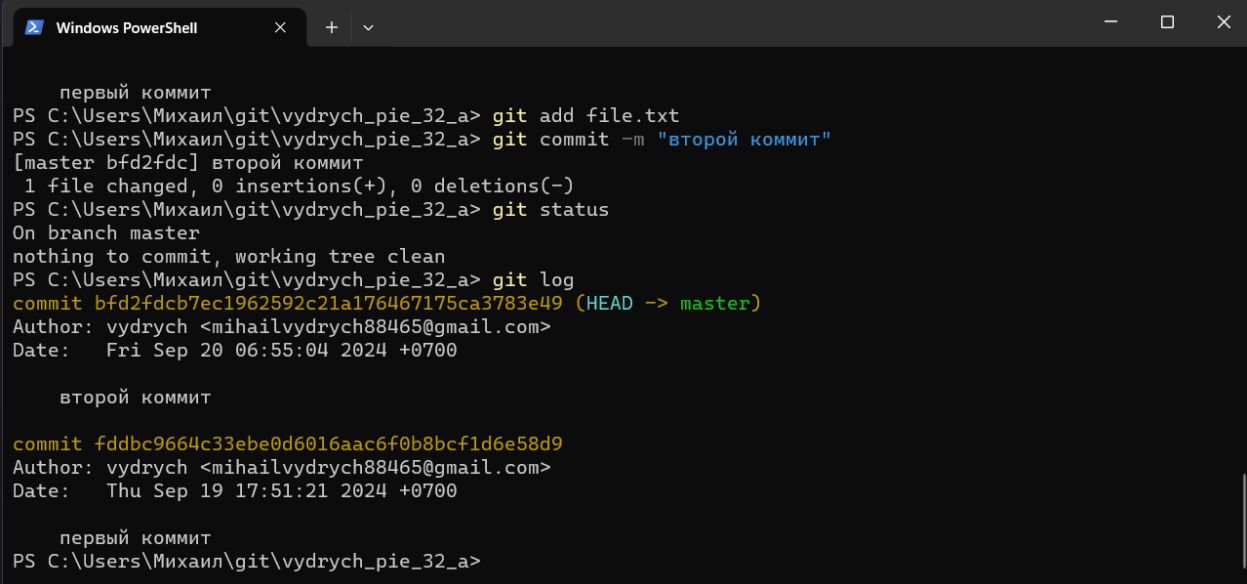
file.txt

```
hello
world
2
```

Строка 3, столбец 2 14 символов

Рисунок 3 – Создание первого коммита

Создаем второй коммит.



```
первый коммит
PS C:\Users\Михаил\git\vydrych_pie_32_a> git add file.txt
PS C:\Users\Михаил\git\vydrych_pie_32_a> git commit -m "второй коммит"
[master bfd2fdc] второй коммит
 1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Михаил\git\vydrych_pie_32_a> git status
On branch master
nothing to commit, working tree clean
PS C:\Users\Михаил\git\vydrych_pie_32_a> git log
commit bfd2fdb7ec1962592c21a176467175ca3783e49 (HEAD -> master)
Author: vydrych <mihailvydrych88465@gmail.com>
Date:   Fri Sep 20 06:55:04 2024 +0700

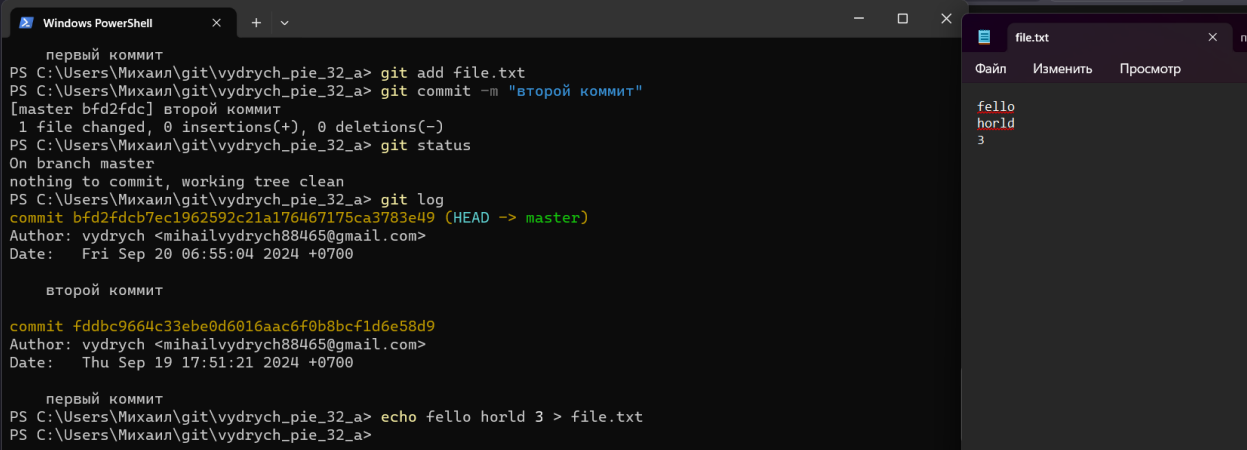
второй коммит

commit fddbc9664c33ebe0d6016aac6f0b8bcf1d6e58d9
Author: vydrych <mihailvydrych88465@gmail.com>
Date:   Thu Sep 19 17:51:21 2024 +0700

первый коммит
PS C:\Users\Михаил\git\vydrych_pie_32_a>
```

Рисунок 4 – Создание второго коммита

Снова вносим изменения в файл командой echo.



```
первый коммит
PS C:\Users\Михаил\git\vydrych_pie_32_a> git add file.txt
PS C:\Users\Михаил\git\vydrych_pie_32_a> git commit -m "второй коммит"
[master bfd2fdc] второй коммит
 1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Михаил\git\vydrych_pie_32_a> git status
On branch master
nothing to commit, working tree clean
PS C:\Users\Михаил\git\vydrych_pie_32_a> git log
commit bfd2fdb7ec1962592c21a176467175ca3783e49 (HEAD -> master)
Author: vydrych <mihailvydrych88465@gmail.com>
Date:   Fri Sep 20 06:55:04 2024 +0700

второй коммит

commit fddbc9664c33ebe0d6016aac6f0b8bcf1d6e58d9
Author: vydrych <mihailvydrych88465@gmail.com>
Date:   Thu Sep 19 17:51:21 2024 +0700

первый коммит
PS C:\Users\Михаил\git\vydrych_pie_32_a> echo fello horld 3 > file.txt
PS C:\Users\Михаил\git\vydrych_pie_32_a>
```

file.txt

```
fello
horld
3
```

Рисунок 5 – запись изменений в файл

И создаем третий коммит

```
Windows PowerShell
Date: Thu Sep 19 17:51:21 2024 +0700

первый коммит
PS C:\Users\Михаил\git\vydrych_pie_32_a> echo fello horld 3 > file.txt
PS C:\Users\Михаил\git\vydrych_pie_32_a> git add file.txt
PS C:\Users\Михаил\git\vydrych_pie_32_a> git commit -m "третий коммит"
[master 07bf680] третий коммит
 1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Михаил\git\vydrych_pie_32_a> git log
commit 07bf68029d8e796936c0ace1730ad5dce5438bb8 (HEAD -> master)
Author: vydrych <mihailvydrych88465@gmail.com>
Date: Fri Sep 20 06:58:17 2024 +0700

третий коммит

commit bfd2fdcb7ec1962592c21a176467175ca3783e49
Author: vydrych <mihailvydrych88465@gmail.com>
Date: Fri Sep 20 06:55:04 2024 +0700

второй коммит

commit fddbc9664c33ebe0d6016aac6f0b8bcf1d6e58d9
Author: vydrych <mihailvydrych88465@gmail.com>
Date: Thu Sep 19 17:51:21 2024 +0700

первый коммит
PS C:\Users\Михаил\git\vydrych_pie_32_a>
```

Рисунок 6 – Создание третьего коммита

Создаем вторую ветку командой git branch.

```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\Users\Михаил\git\vydrych_pie_32_a> git branch second
PS C:\Users\Михаил\git\vydrych_pie_32_a> git chechout second
git: 'chechout' is not a git command. See 'git --help'.

The most similar command is
checkout
PS C:\Users\Михаил\git\vydrych_pie_32_a> git checkout second
Switched to branch 'second'
PS C:\Users\Михаил\git\vydrych_pie_32_a> |
```

Рисунок 7 – Создание второй ветки

Вносим изменения в файл и добавляем его в коммит

```
Windows PowerShell
PS C:\Users\Михаил\git\vydrych_pie_32_a> echo second branch 1 > file.txt
PS C:\Users\Михаил\git\vydrych_pie_32_a> git add file.txt
PS C:\Users\Михаил\git\vydrych_pie_32_a> git commit -m 'новая ветка первый коммит'
[second 57b5fae] новая ветка первый коммит
1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Михаил\git\vydrych_pie_32_a> git log
commit 57b5faefd4aa513f29b0a7f4f3bc94ef4b088a93 (HEAD -> second)
Author: vydrych <mihailvydrych88465@gmail.com>
Date:   Fri Sep 20 22:10:54 2024 +0700

    новая ветка первый коммит

commit 07bf68029d8e796936c0ace1730ad5dce5438bb8 (master)
Author: vydrych <mihailvydrych88465@gmail.com>
Date:   Fri Sep 20 06:58:17 2024 +0700

    третий коммит

commit bfd2fdbc7ec1962592c21a176467175ca3783e49
Author: vydrych <mihailvydrych88465@gmail.com>
Date:   Fri Sep 20 06:55:04 2024 +0700

    второй коммит

commit fddbc9664c33ebe0d6016aac6f0b8bcf1d6e58d9
Author: vydrych <mihailvydrych88465@gmail.com>
Date:   Thu Sep 19 17:51:21 2024 +0700

    первый коммит
PS C:\Users\Михаил\git\vydrych_pie_32_a> |
```

Рисунок 8 – Создание первого коммита во второй ветке

Создаем второй коммит в новой ветке. Сначала так же вносим изменения в файл.

Делаем слияние веток командой merge.

```
Windows PowerShell
PS C:\Users\Михаил\git\vydrych_pie_32_a> git branch
* master
  second
PS C:\Users\Михаил\git\vydrych_pie_32_a> git checkout master
Switched to branch 'master'
PS C:\Users\Михаил\git\vydrych_pie_32_a> git log
commit 07bf68029d8e796936c0ace1730ad5dce5438bb8 (HEAD -> master)
Author: vydrych <mihailvydrych88465@gmail.com>
Date:   Fri Sep 20 06:58:17 2024 +0700

    третий коммит

commit bfd2fdbc7ec1962592c21a176467175ca3783e49
Author: vydrych <mihailvydrych88465@gmail.com>
Date:   Fri Sep 20 06:55:04 2024 +0700

    второй коммит

commit fddbc9664c33ebe0d6016aac6f0b8bcf1d6e58d9
Author: vydrych <mihailvydrych88465@gmail.com>
Date:   Thu Sep 19 17:51:21 2024 +0700

    первый коммит
PS C:\Users\Михаил\git\vydrych_pie_32_a> git merge second
Updating 07bf680..3fbc9ac
Fast-forward
 file.txt | Bin 36 -> 40 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Михаил\git\vydrych_pie_32_a> |
```

Рисунок 9 – Слияние веток

Результат

```
Windows PowerShell
file.txt | Bin 36 -> 40 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Михаил\git\vydrych_pie_32_a> git log
commit 3f9e9acdb3bc00d52e56a5b2cefb24081e3ecd4 (HEAD -> master, second)
Author: vydrych <mihailvydrych88465@gmail.com>
Date:   Fri Sep 20 22:37:41 2024 +0700

    новая ветка второй коммит

commit 57b5faefd4aa513f29b0a7f4f3bc94ef4b088a93
Author: vydrych <mihailvydrych88465@gmail.com>
Date:   Fri Sep 20 22:10:54 2024 +0700

    новая ветка первый коммит

commit 07bf68029d8e796936c0ace1730ad5dce5438bb8
Author: vydrych <mihailvydrych88465@gmail.com>
Date:   Fri Sep 20 06:58:17 2024 +0700

    третий коммит

commit bfd2fdcb7ec1962592c21a176467175ca3783e49
Author: vydrych <mihailvydrych88465@gmail.com>
Date:   Fri Sep 20 06:55:04 2024 +0700

    второй коммит

commit fddbc9664c33ebe0d6016aac6f0b8bcf1d6e58d9
Author: vydrych <mihailvydrych88465@gmail.com>
Date:   Thu Sep 19 17:51:21 2024 +0700

    первый коммит
PS C:\Users\Михаил\git\vydrych_pie_32_a> |
```

Рисунок 10 – результат слияния

Скачиваем гитхаб десктоп, входим в аккаунт и создаем сначала локальный репозиторий.

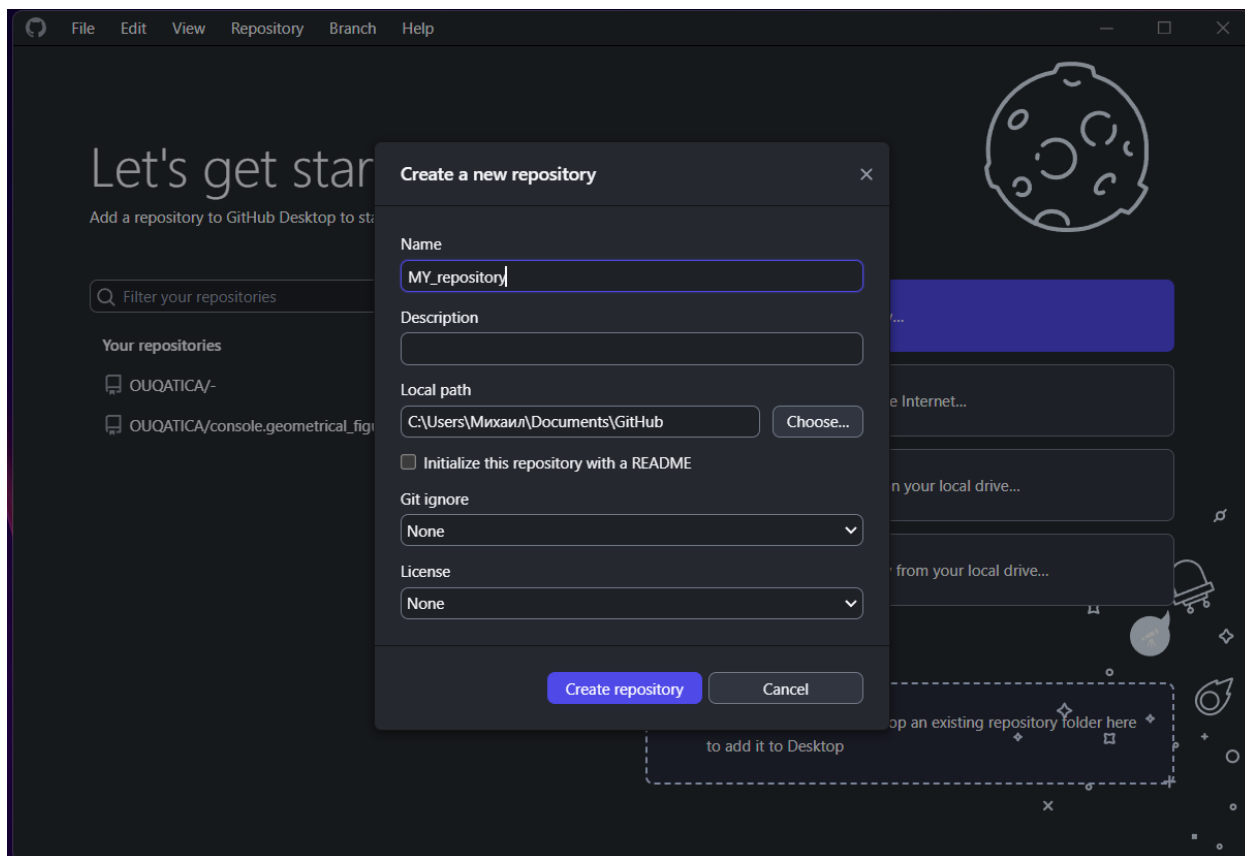


Рисунок 11 – Создание репозитория

Нажимаем Publish repository и как бы отправляем его на сервер гитхаба.

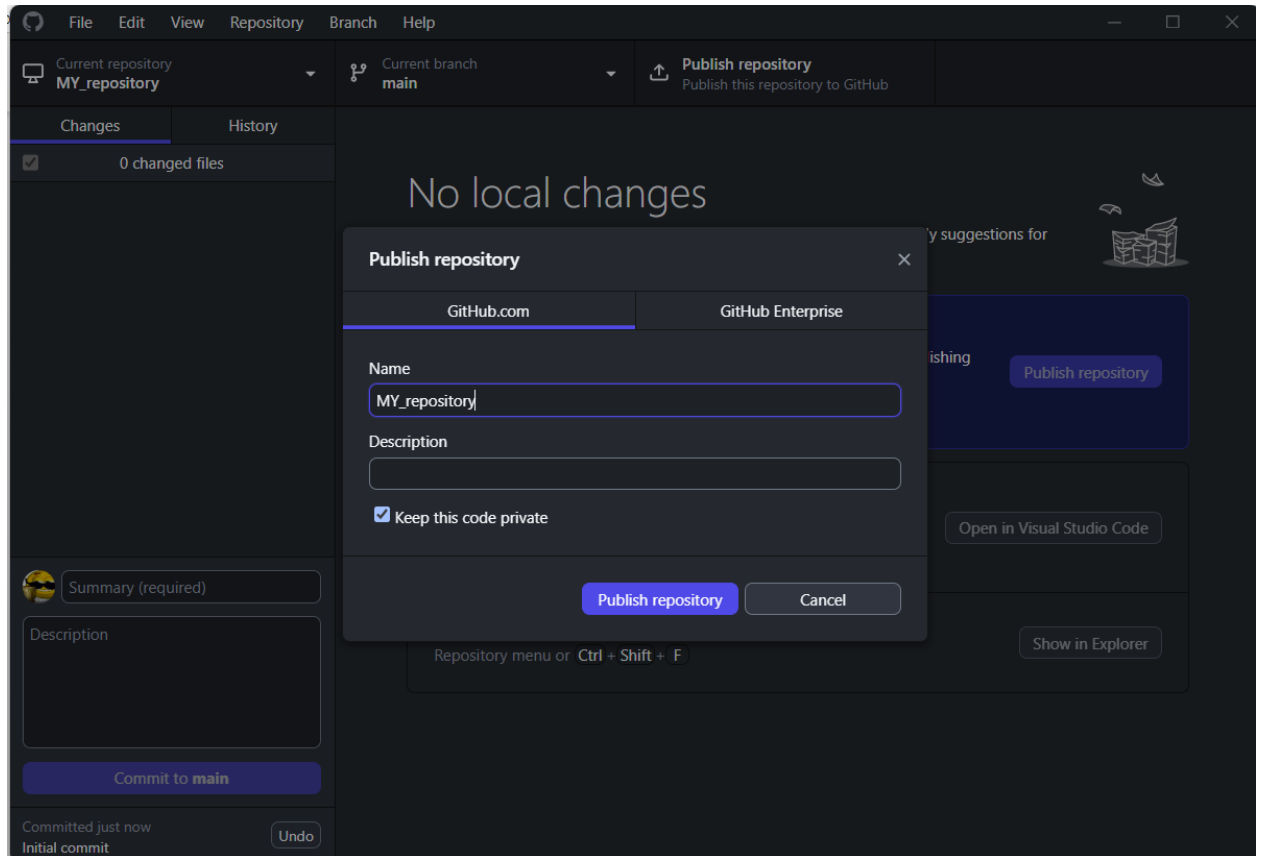


Рисунок 12 – создание удаленного репозитория

Он больше не локальный. Теперь открываем репозиторий в проводнике и создаем файл. Записываем в него любую белеберду и файл появляется в основной ветке. Изменения так же видны.

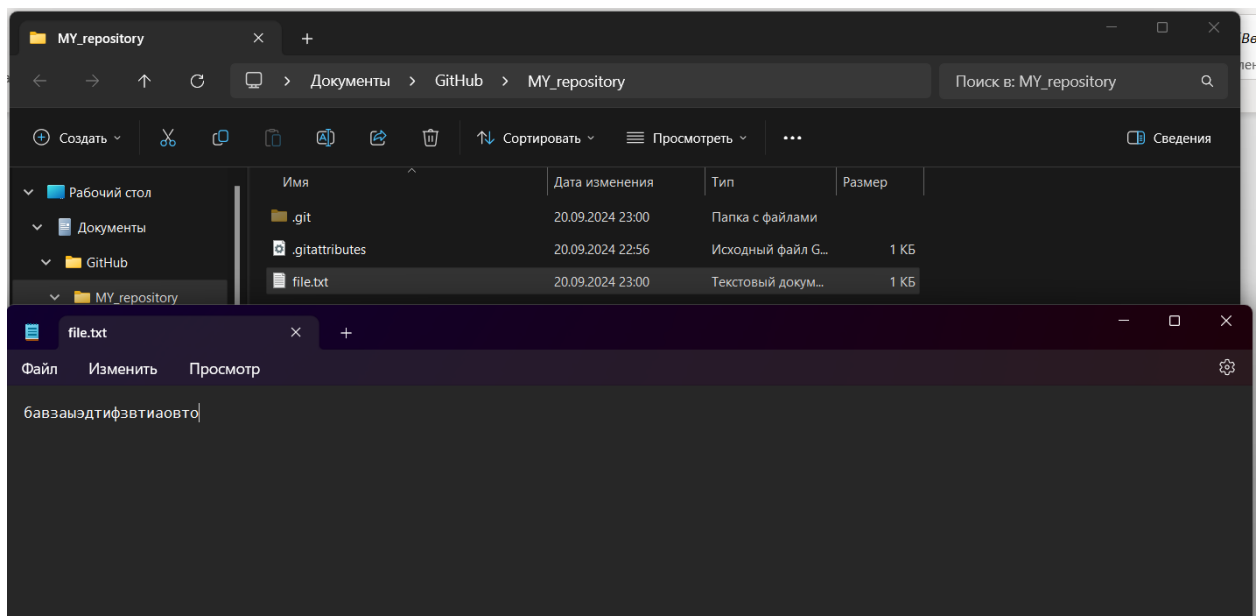


Рисунок 13 – создание файла

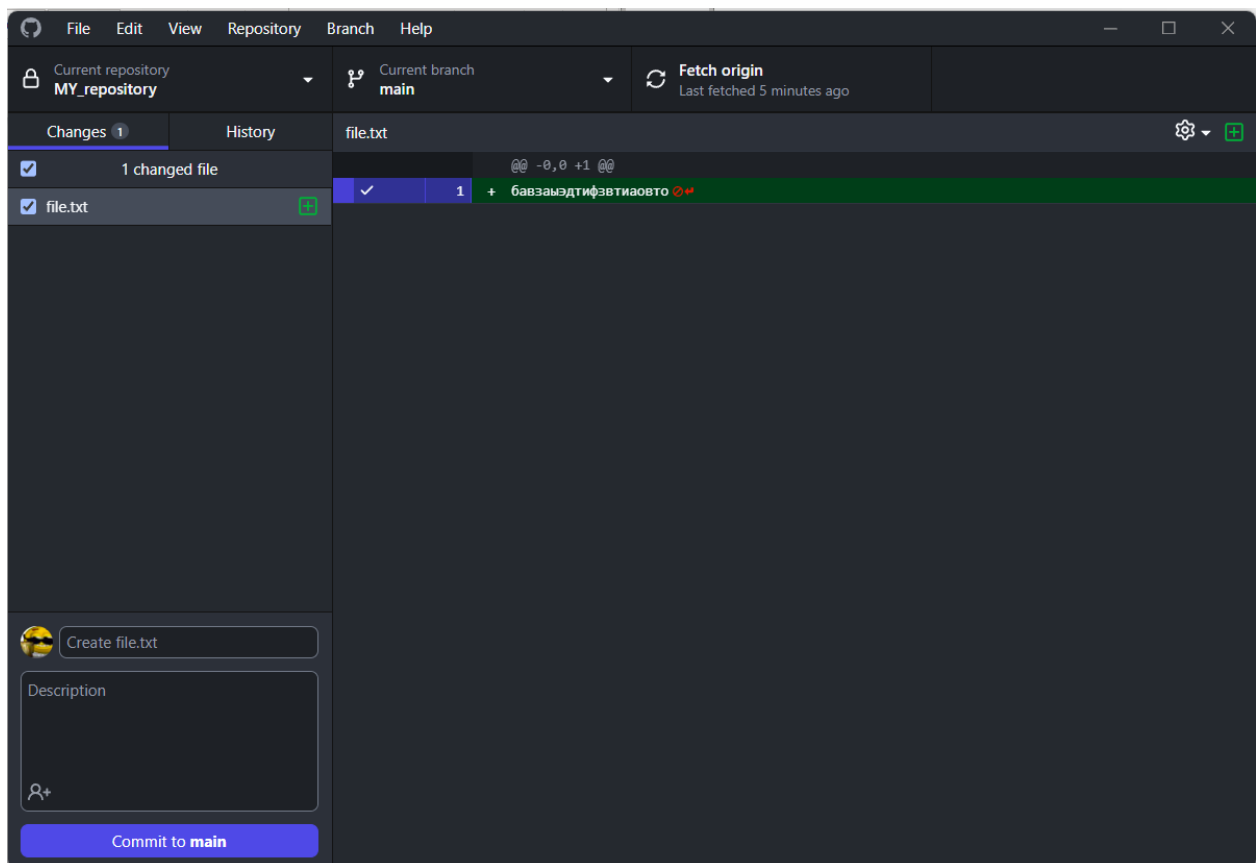


Рисунок 14 – проверка видимости изменений

Нажимаем commit to main и коммит создан. В истории смотрим, что он действительно появился.

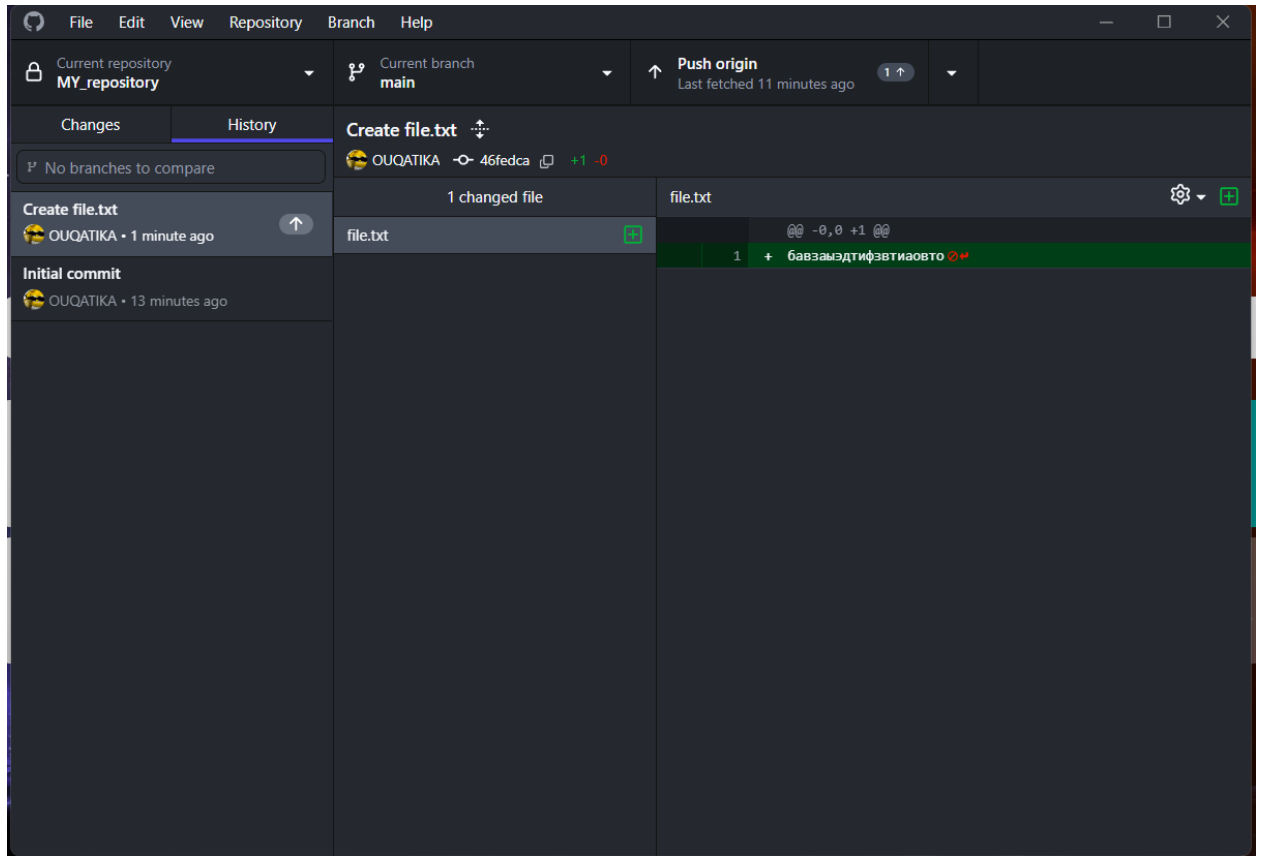


Рисунок 15 – Создание первого коммита

Вносим изменения в файл и снова видим их в программе. Делаем КОММИТ.

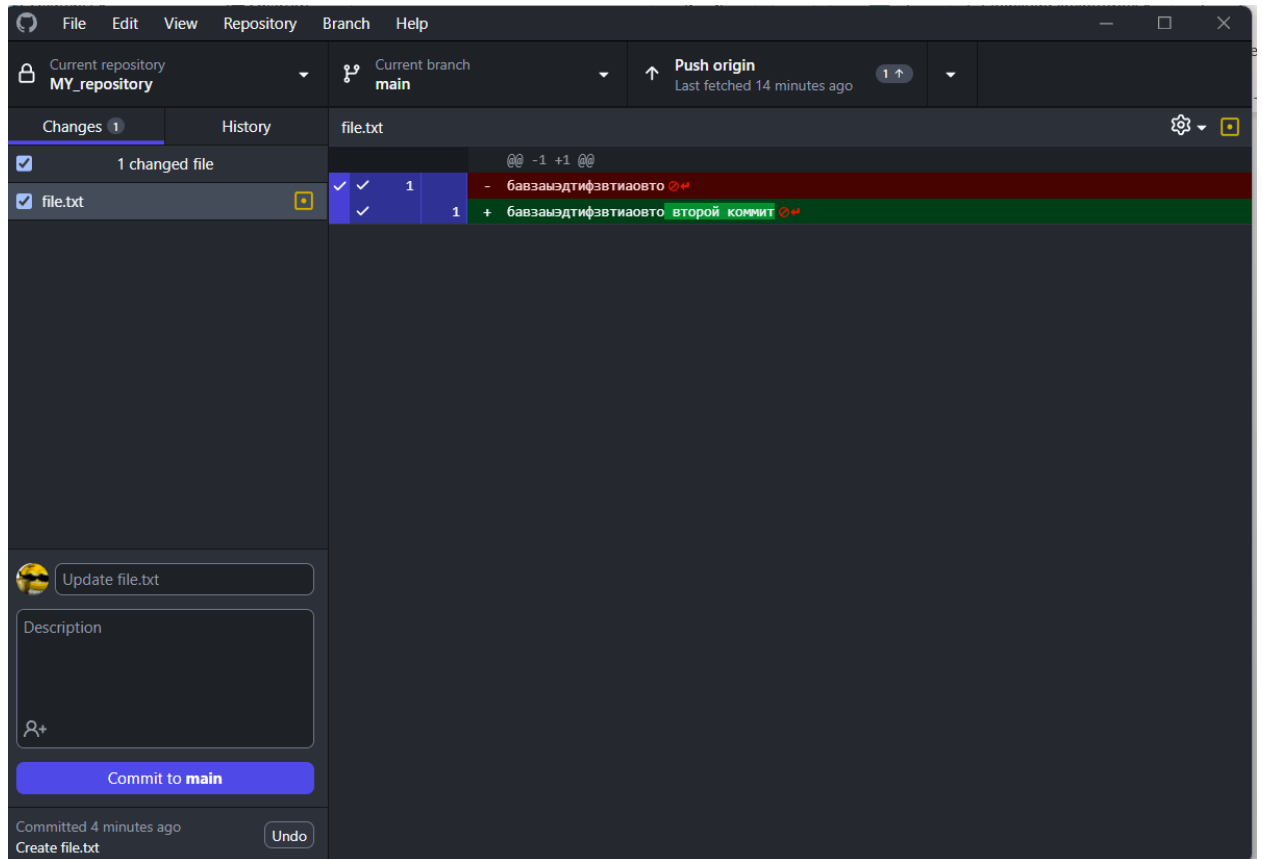


Рисунок 16 – Создание второго коммита

В третий раз изменяем файл и создаем коммит.

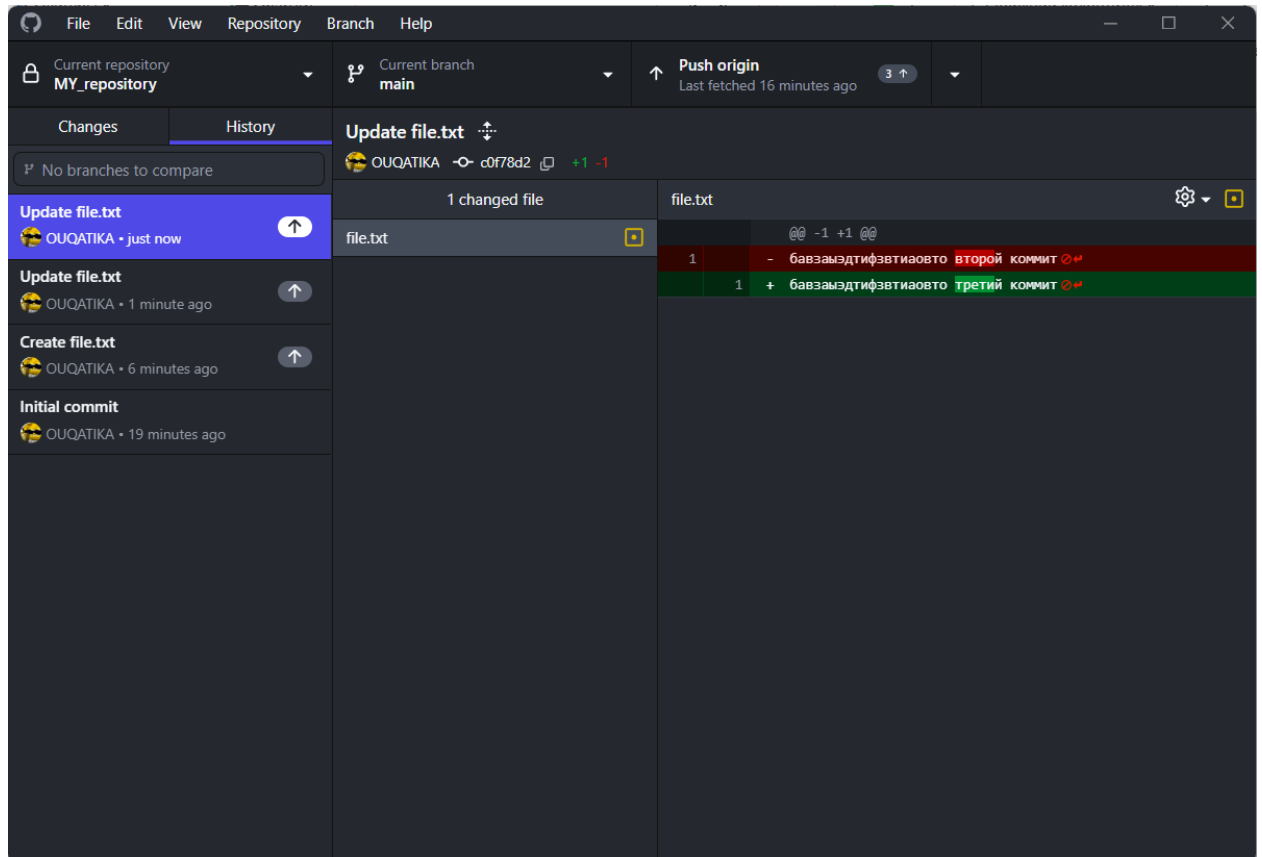


Рисунок 17 – Создание третьего коммита

Создаем вторую ветку и называем ее second_branch.

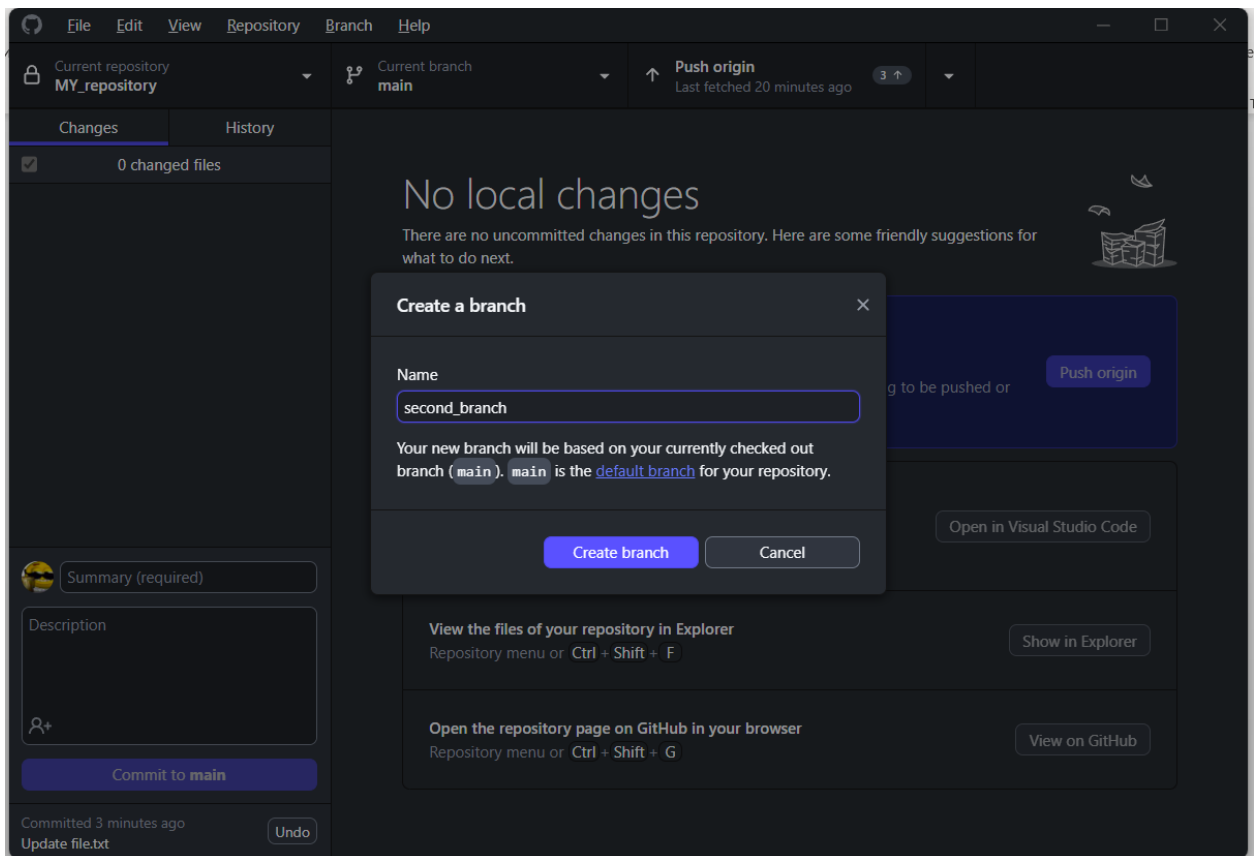


Рисунок 18 – Создание новой ветки

Эти два коммита созданы уже во второй ветке.

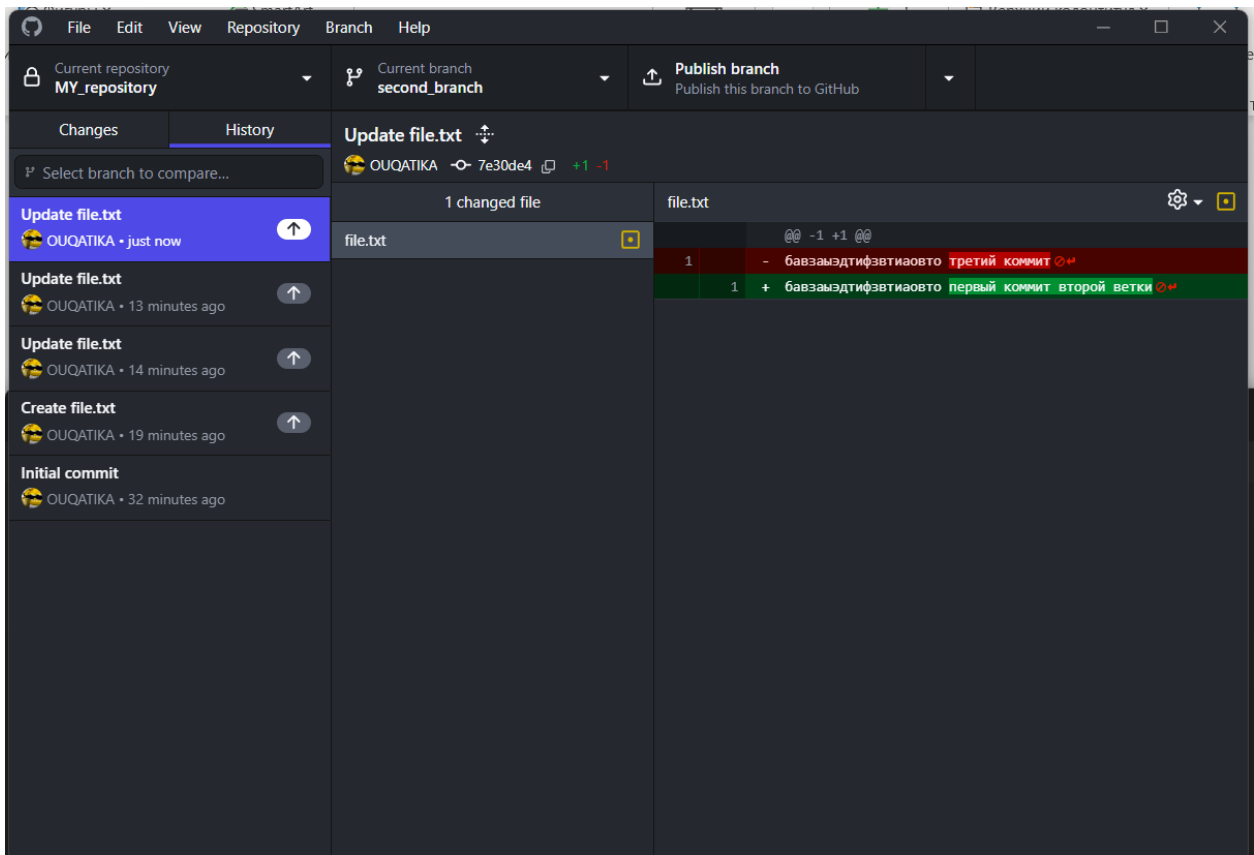


Рисунок 19 – Создание первого коммита второй ветки

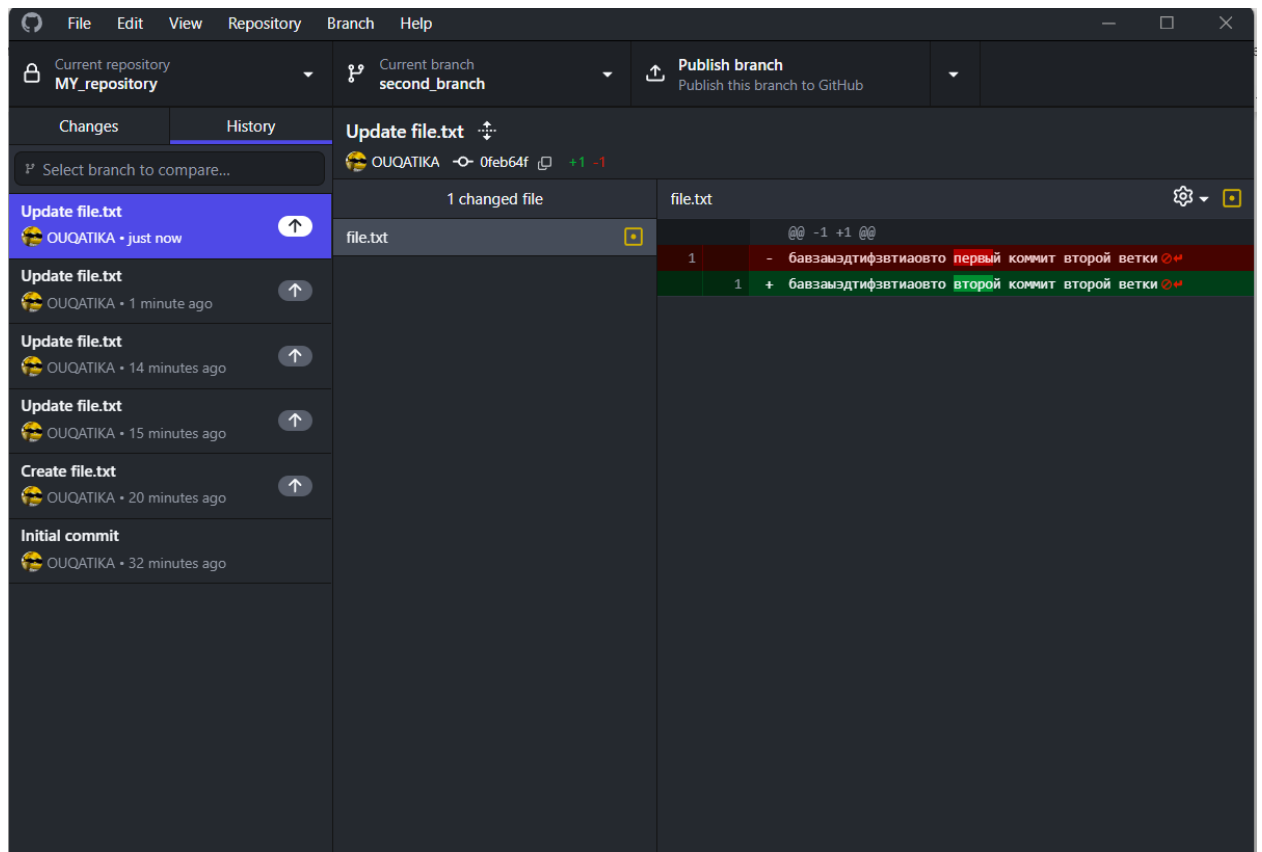


Рисунок 20 – Создание второго коммита новой ветки

Слияние веток делаем через Choose a branch to merge into main.

Выбираем вторую ветку.

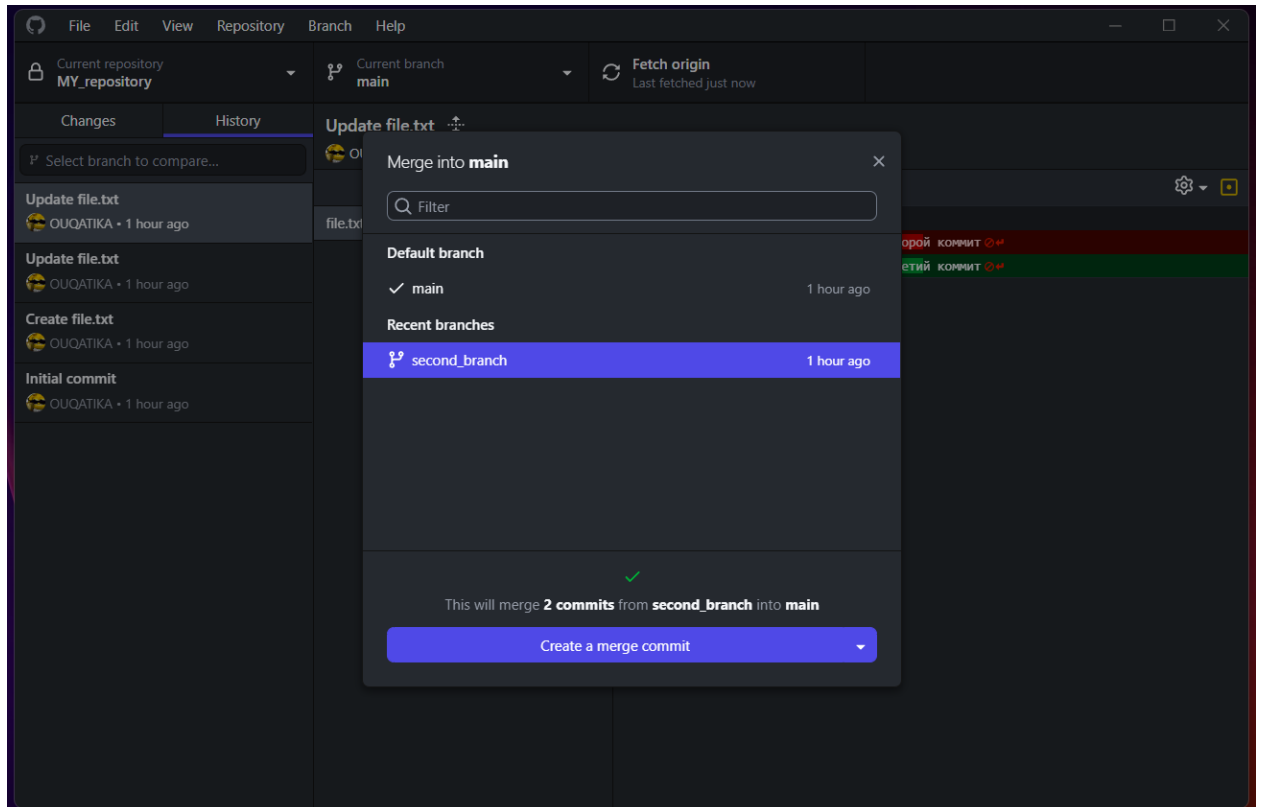


Рисунок 21 – слияние веток

После этого синхронизируем обе ветки с удаленным репозиторием, нажимая на Publish branch.

Древо коммитов:

В локальном репозитории я сделал с помощью git kraken

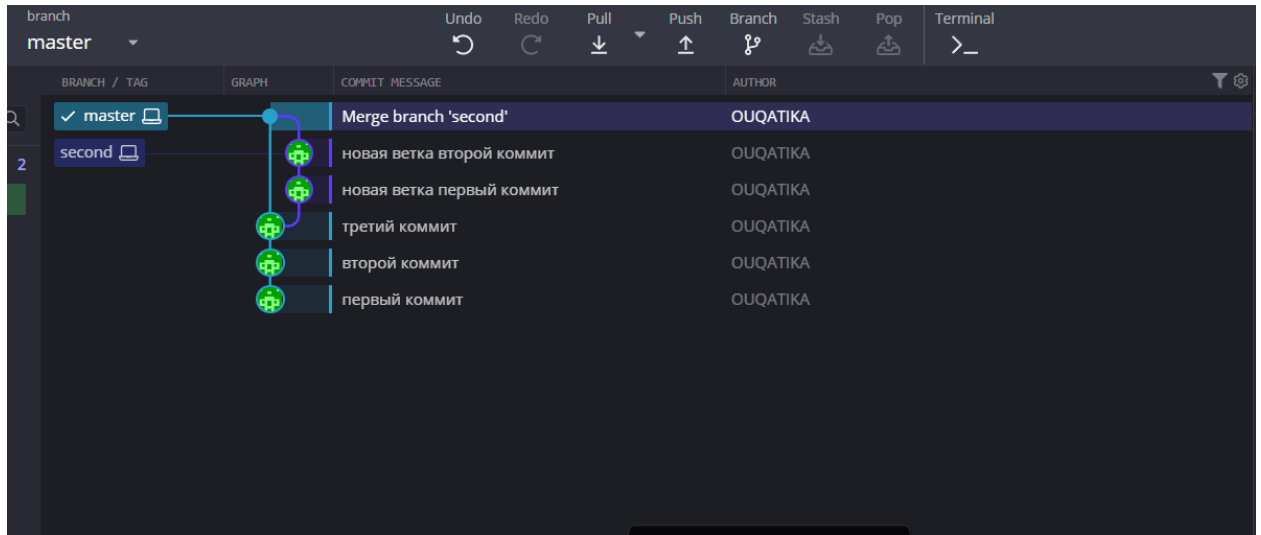


Рисунок 21 – древо коммитов в Git Kraken

В Github древо коммитов находится в Insights/Network.

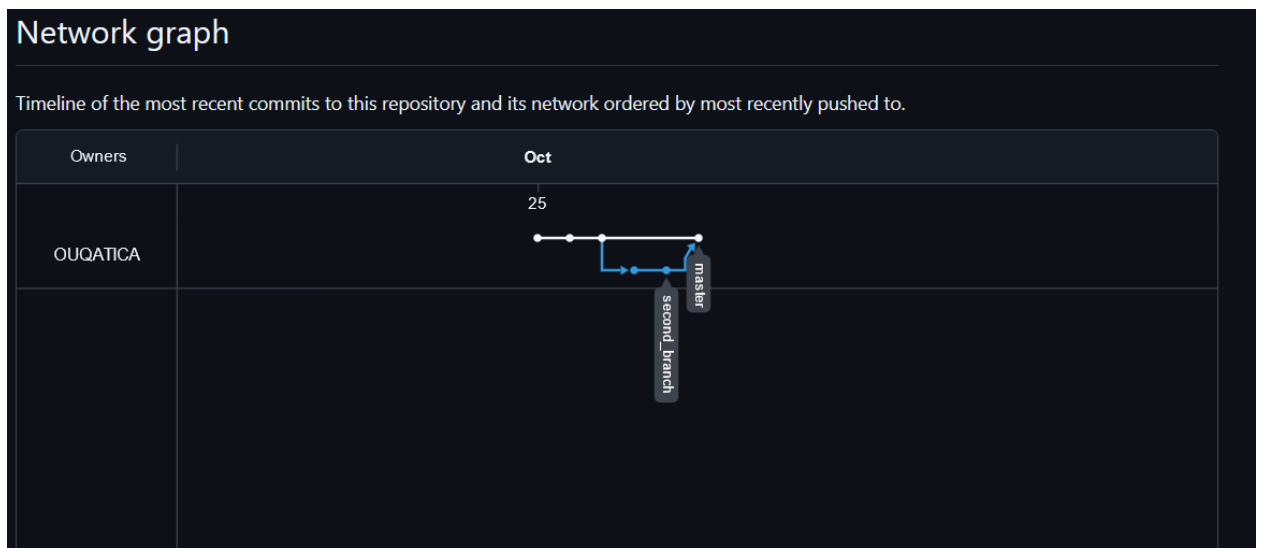


Рисунок 22 – древо коммитов в github

Вывод:

В ходе выполнения этой работы я приобрел ценные навыки, связанные с созданием и управлением репозиториями в системе контроля версий Git. Я изучил основные команды Git, такие как `git init`, `git branch`, `git commit` и другие.

Кроме того, я освоил использование GitHub Desktop и Git Kraken, которые значительно упрощают процесс взаимодействия с удаленными репозиториями. Эти инструменты предоставляют удобный графический интерфейс для выполнения таких операций, как создание веток, слияние веток, просмотр древа коммитов. Это сделало работу с Git более интуитивно понятной.

Также я научился выводить древо коммитов, что дало мне возможность визуализировать историю изменений. Используя `git kraken` и `github`, я смог увидеть структуру веток и коммитов, которая помогает лучше понять развитие “проекта” (в кавычках, потому что в ходе выполнения этой лабораторной никакого проекта не было) и как различные изменения связаны друг с другом.

В целом, этот опыт не только дал мне знания о Git и углубил знание GitHub, но также позволит мне в будущем при разработке каких-то относительно больших личных, учебных, или рабочих проектов более уверенно вносить изменения и не волноваться по поводу сохранения работоспособности кода, что является важным навыком для любого разработчика.

https://github.com/OUQATICA/MY_repository