

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего  
образования  
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ  
УНИВЕРСИТЕТ»**

**Кафедра  
инфокоммуникаций  
Институт цифрового  
развития**

**ОТЧЁТ**  
**по лабораторной работе №1.3**  
Дисциплина: «Основы программной инженерии»  
Тема: «Основы ветвления GIT»

Выполнила:  
студентка 2 курса  
группы Пиж-б-о-21-1  
Джолдошова Мээрим  
Бекболотовна

Ставрополь 2022

**Цель:** исследование базовых возможностей по работе с локальными и удаленными ветками GIT

## Выполнение работы

1. Был создан репозиторий с использованием лицензии MIT и файлами gitignore для выбранного языка программирования C++. Далее он был клонирован по ноутбук.

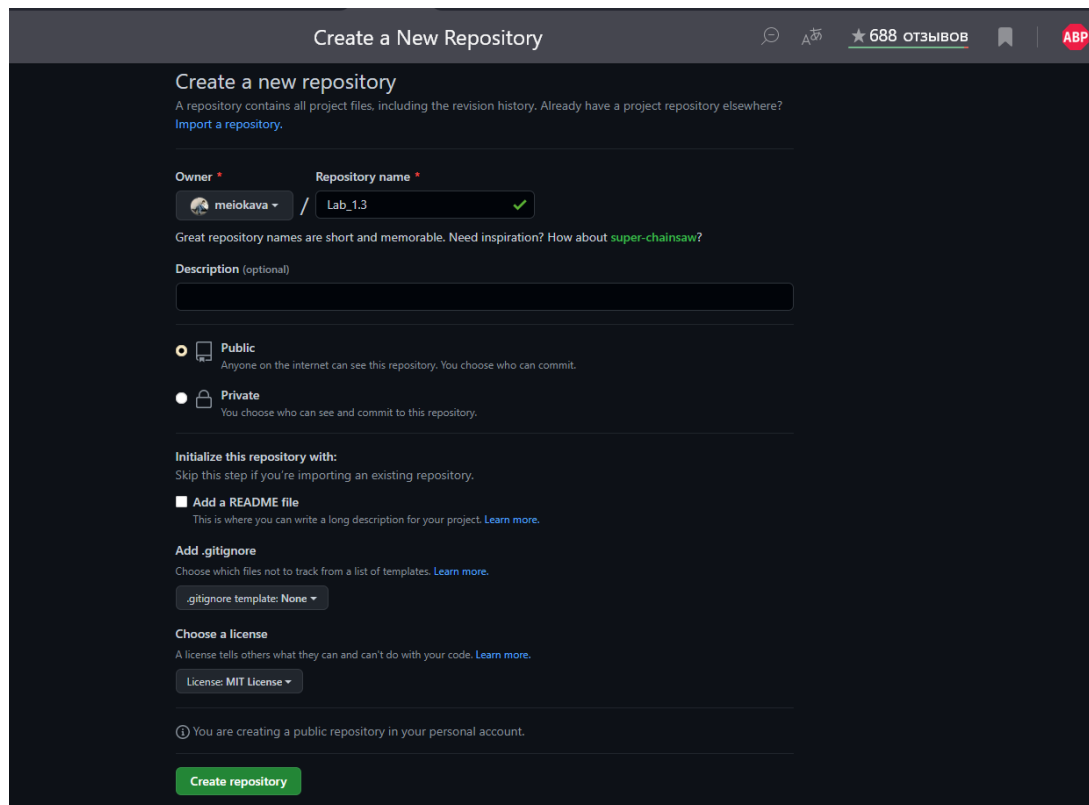


Рисунок 1.1 – Создание репозитория

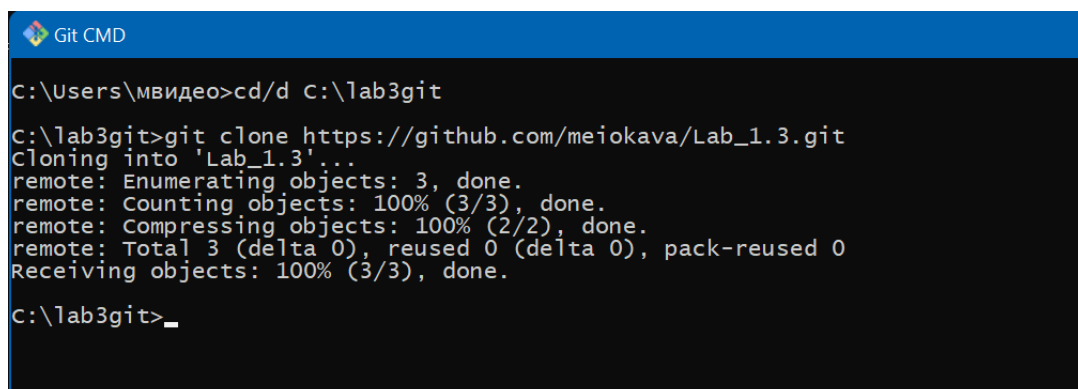


Рисунок 1.2 – Клонирование репозитория

2. Создание трех текстовых файлов

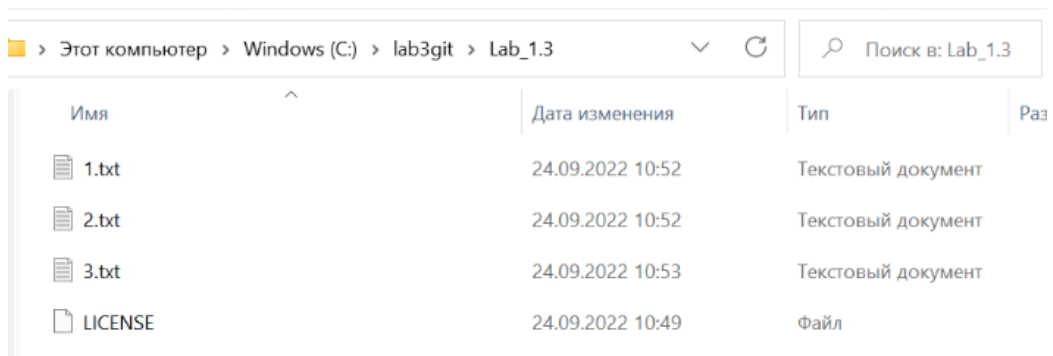


Рисунок 2.1 – Создание трех текстовых файлов

```
C:\Users\мвидео>cd/d C:\lab3git
C:\lab3git>git clone https://github.com/meiokava/Lab_1.3.git
Cloning into 'Lab_1.3'...
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:\lab3git>git add 1.txt
fatal: not a git repository (or any of the parent directories): .git
C:\lab3git>cd/d C:\lab3git\Lab_1.3
C:\lab3git\Lab_1.3>git add 1.txt
C:\lab3git\Lab_1.3>git commit -m "add 1.txt file"
[main 288b865] add 1.txt file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt
C:\lab3git\Lab_1.3>_
```

Рисунок 2.2 – Коммит с комментарием add 1.txt file

```
C:\lab3git\Lab_1.3>git add .
C:\lab3git\Lab_1.3>git commit --amend -m "add 2.txt and 3.txt"
[main 24d3f0c] add 2.txt and 3.txt
Date: Sat Sep 24 10:57:08 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:\lab3git\Lab_1.3>_
```

Рисунок 2.3 – индексация второго и третьего файла

3. Была создана новая ветка `my_first_branch`, был осуществлен переход на нее и создан новый файл в нем `in_branch.txt`, были закоммичены изменения. После этого произошел возврат на ветку `master`, и переход на ветку `new_branch`, были осуществлены изменения в файле `1.txt` и закоммичены изменения.

```
C:\lab3git\Lab_1.3>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\lab3git\Lab_1.3>git branch -a
* main
  my_first_branch
remotes/origin/HEAD -> origin/main
remotes/origin/main

C:\lab3git\Lab_1.3>git
```

Рисунок 3.1 – Отображение веток, а также текущей ветки

```
C:\lab3git\Lab_1.3>git checkout my_first_branch
Switched to branch 'my_first_branch'

C:\lab3git\Lab_1.3>
```

Рисунок 3.2 – Переход на ветку my\_first\_branch

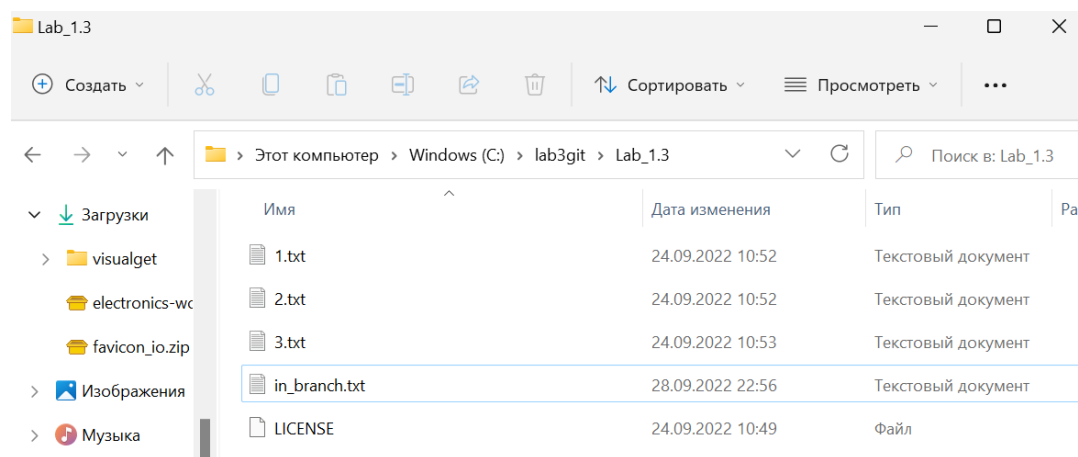


Рисунок 3.3 – Добавление нового текстового файла в репозиторий

```
C:\lab3git\Lab_1.3>git branch -a
main
* my_first_branch
remotes/origin/HEAD -> origin/main
remotes/origin/main

C:\lab3git\Lab_1.3>git add .

C:\lab3git\Lab_1.3>git commit -m "new addition in_branch.txt"
[my_first_branch 4f74561] new addition in_branch.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 in_branch.txt

C:\lab3git\Lab_1.3>
```

Рисунок 3.4 – Коммит сделанных изменений

```

C:\lab3git\Lab_1.3>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:\lab3git\Lab_1.3>git checkout -b new_branch
Switched to a new branch 'new_branch'

C:\lab3git\Lab_1.3>

```

Рисунок 3.5 – Переход на главную ветвь и создание с переходом на новую ветку

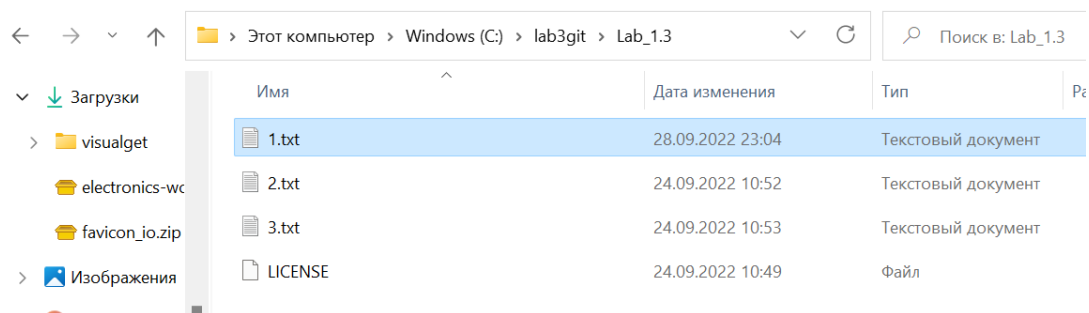


Рисунок 3.6 – В новой ветви нет добавленного файла в ветви my\_first\_branch

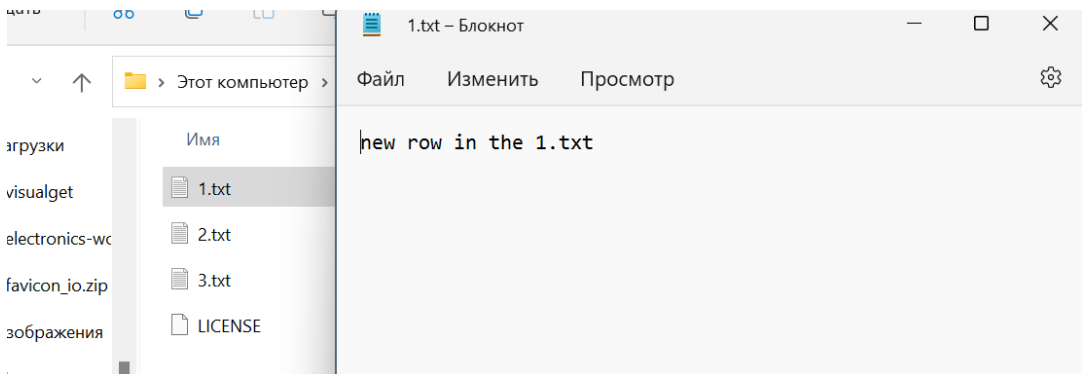


Рисунок 3.7 – Изменения в файле 1.txt

```

C:\lab3git\Lab_1.3>git commit -am "changes in 1.txt"
[new_branch 0fb42f7] changes in 1.txt
1 file changed, 1 insertion(+)

C:\lab3git\Lab_1.3>

```

Рисунок 3.8 – Коммит изменений

4. Был осуществлен переход на ветку main также были слиты ветки

main и my\_first\_branch. После были удалены все ветки кроме main.

```
C:\lab3git\Lab_1.3>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:\lab3git\Lab_1.3>git merge my_first_branch
Updating 24d3f0c..4f74561
Fast-forward
 in_branch.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 in_branch.txt
```

Рисунок 4.1 – Переход на ветку main и слияние ее с веткой my\_first\_branch

```
C:\lab3git\Lab_1.3>git branch
* main
  my_first_branch
  new_branch

C:\lab3git\Lab_1.3>git merge new_branch
Merge made by the 'ort' strategy.
 1.txt | 1 +
 1 file changed, 1 insertion(+)
```

Рисунок 4.2 – Слияние ветки main и ветки new\_branch

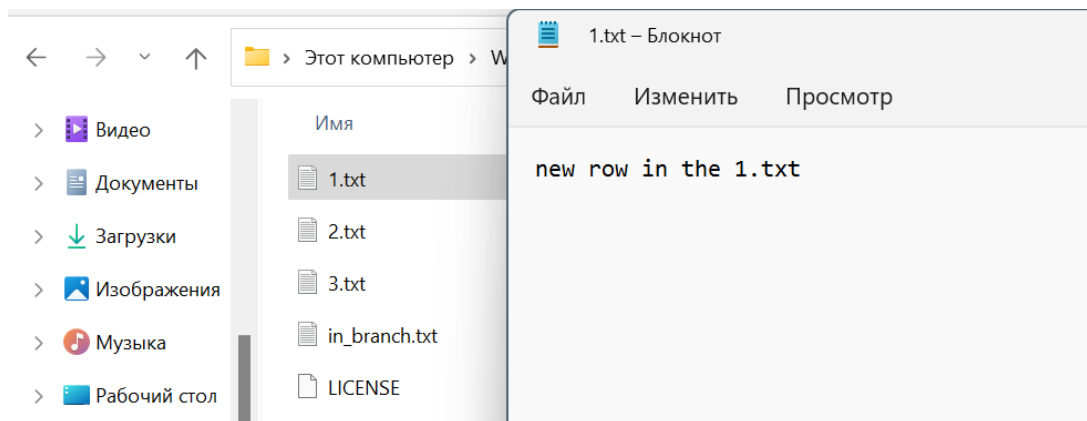


Рисунок 4.3 – Вид репозитория на ветви main

```
C:\lab3git\Lab_1.3>git branch -d my_first_branch
Deleted branch my_first_branch (was 4f74561).

C:\lab3git\Lab_1.3>git branch -d new_branch
Deleted branch new_branch (was 0fb42f7).

C:\lab3git\Lab_1.3>
```

Рисунок 4.4 – Удаление ветвей my\_first\_branch и new\_branch

5. Создание веток branch\_1 и branch\_2. Переход на ветку branch\_1 и изменение файлов 1.txt, 3.txt. Затем коммит изменений. Осуществление перехода на ветку branch\_2 и изменения файлов 1.txt, 3.txt, после чего фиксация коммитов изменения. После слияние изменения ветки branch\_2 в ветку branch\_2.

```
git: branch_1 is not a git command. See 'git help'.
```

```
C:\lab3git\Lab_1.3>git branch branch_1
C:\lab3git\Lab_1.3>git branch branch_2
C:\lab3git\Lab_1.3>git status
On branch main
Your branch is ahead of 'origin/main' by 4 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\lab3git\Lab_1.3>git branch -a
branch_1
branch_2
* main
remotes/origin/HEAD -> origin/main
remotes/origin/main
C:\lab3git\Lab_1.3>
```

Рисунок 5.1 – Создание веток branch\_1 и branch\_2

```
C:\lab3git\Lab_1.3>git checkout branch_1
Switched to branch 'branch_1'

C:\lab3git\Lab_1.3>
```

Рисунок 5.2 – Переход на ветвь branch\_1

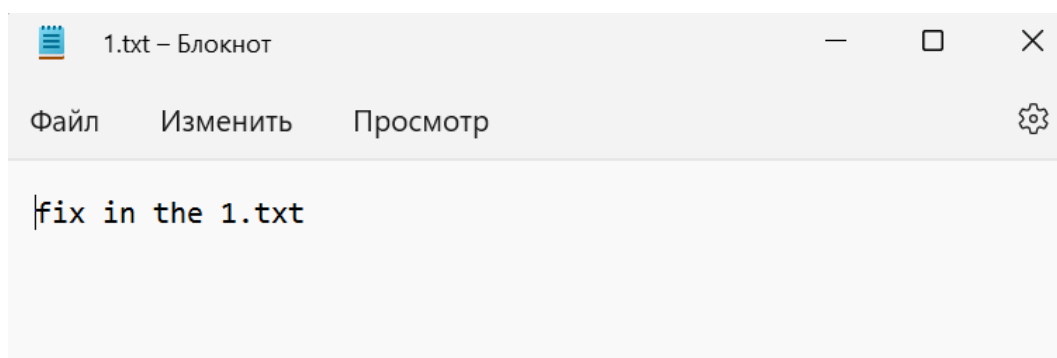


Рисунок 5.3 – Изменения в файле 1.txt

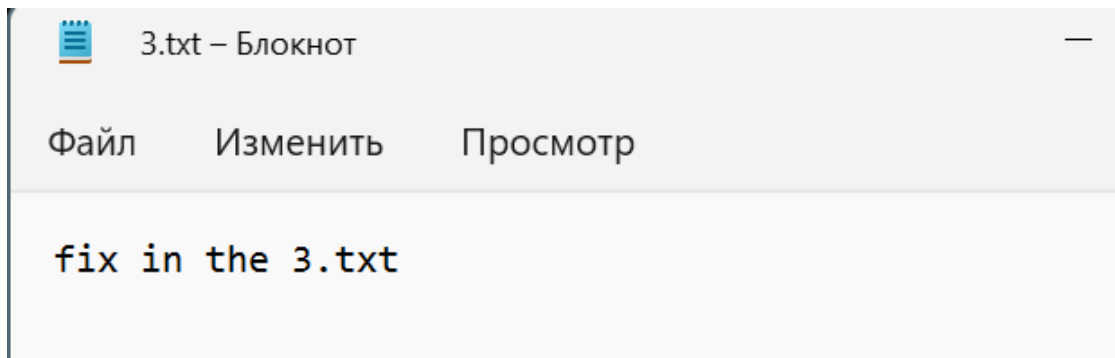


Рисунок 5.4 – Изменения в файле 3.txt

```
C:\lab3git\Lab_1.3>git add .
C:\lab3git\Lab_1.3>git commit -m "changes in the 1.txt and 3.txt in branch_1"
[branch_1 a4c90f0] changes in the 1.txt and 3.txt in branch_1
2 files changed, 2 insertions(+), 1 deletion(-)
C:\lab3git\Lab_1.3>
```

Рисунок 5.5 – Коммит изменений

```
C:\lab3git\Lab_1.3>git checkout branch_2
Switched to branch 'branch_2'
C:\lab3git\Lab_1.3>_
```

Рисунок 5.6 – Переход на ветку branch\_2

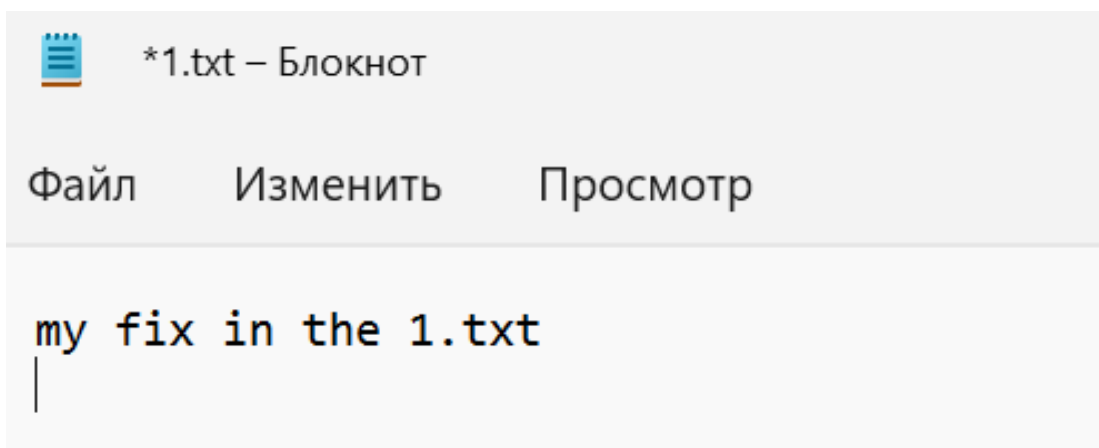


Рисунок 5.7 – Изменения в файле 1.txt



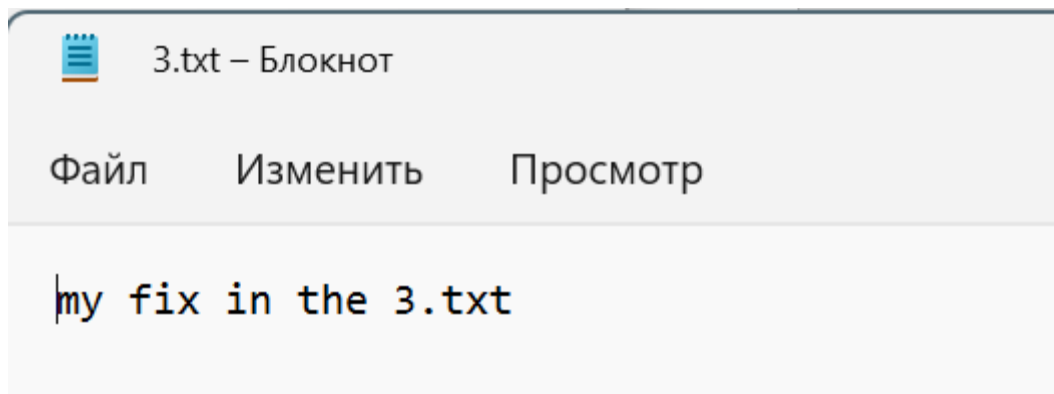


Рисунок 5.8 – Изменение в файле 3.txt

```
C:\lab3git\Lab_1.3>git commit -am "fixation 1.txt and 3.txt in branch_2"
[branch_2 258c23f] fixation 1.txt and 3.txt in branch_2
 2 files changed, 2 insertions(+), 1 deletion(-)
C:\lab3git\Lab_1.3>
```

Рисунок 5.9 – Коммит изменений

## 6. Выполнение слияния веток и устранение конфликтов

```
C:\lab3git\Lab_1.3>git checkout branch_1
Switched to branch 'branch_1'

C:\lab3git\Lab_1.3>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:\lab3git\Lab_1.3>_
```

Рисунок 6.1 – Слияние веток

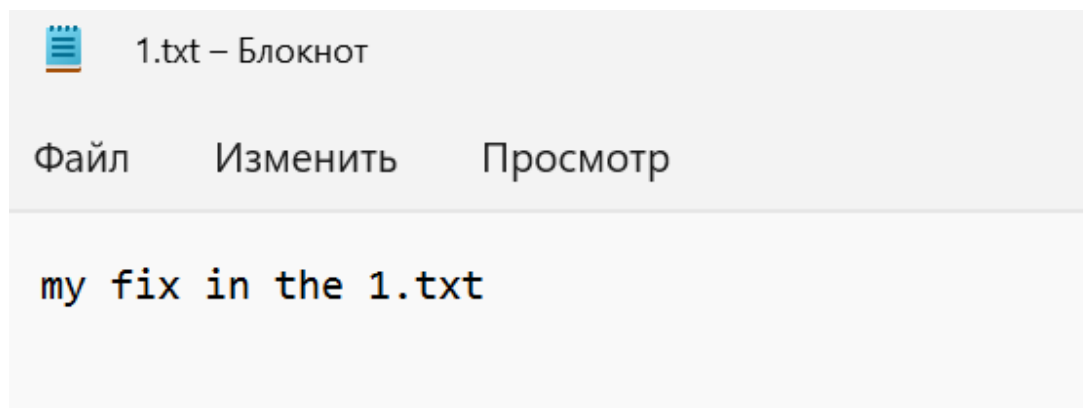


Рисунок 6.2 – Решение конфликта вручную

```

C:\lab3git\Lab_1.3>git add 1.txt
C:\lab3git\Lab_1.3>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

C:\lab3git\Lab_1.3>_

```

Рисунок 6.3 – Добавление индекса 1.txt для решения конфликта

```

C:\lab3git\Lab_1.3>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

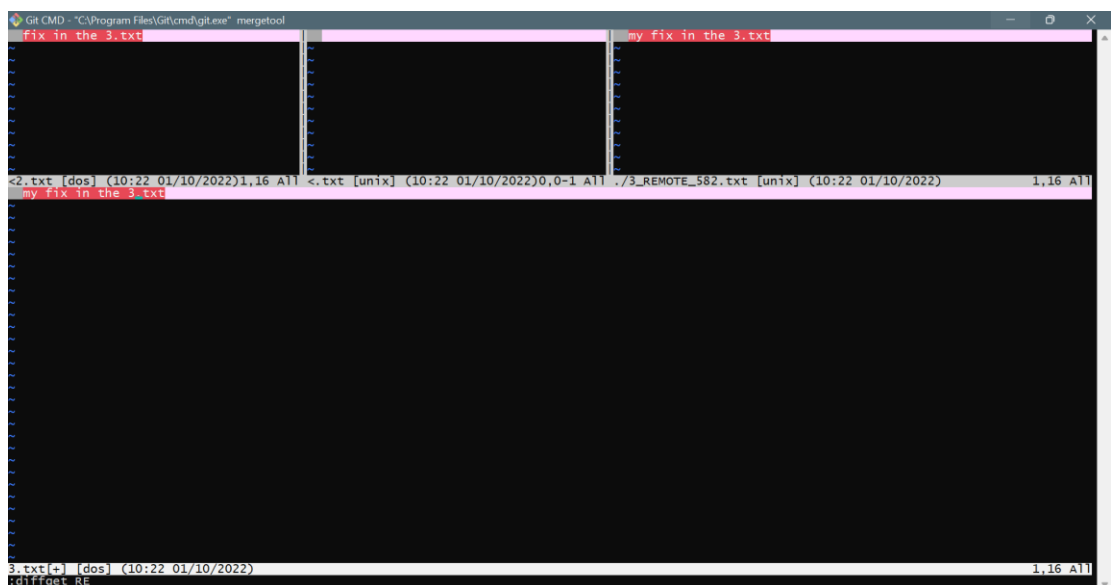


Рисунок 6.5 – Команда diffg RE

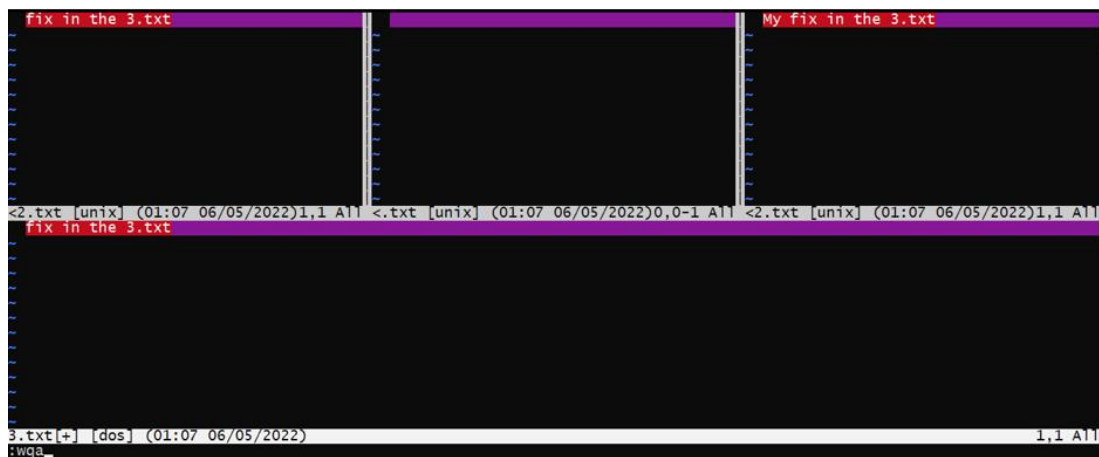


Рисунок 6.6 – Команда wga



Рисунок 6.7 – Осуществление пуша ветки на удаленный сервер

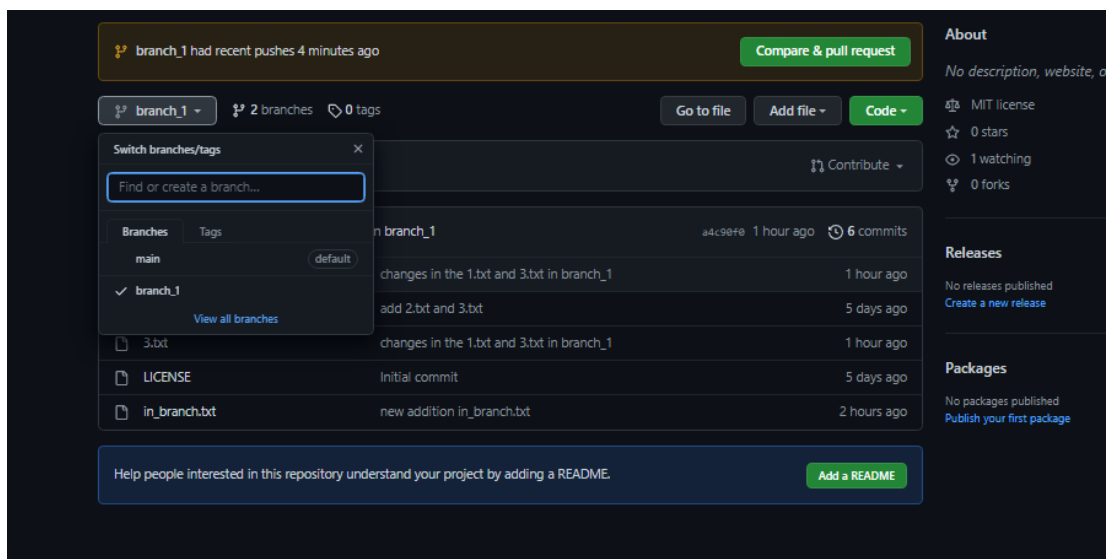


Рисунок 6.8 – Изменения на удаленном сервере

7. Было создано средствами GitHub удаленная ветка branch\_3, также

была создана ветка в локальной репозитории для отслеживания удаленной ветки branch\_3. Был осуществлен переход на ветку branch\_3 и была добавлена строка `the final fantasy in the 4.txt` file»

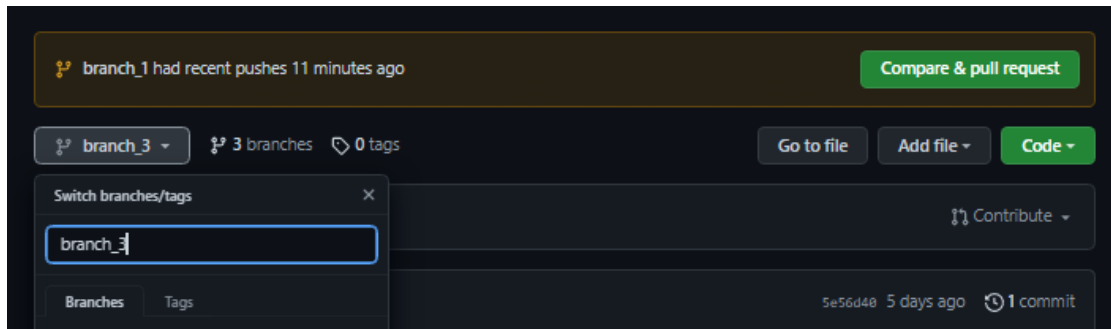


Рисунок 7.1 – Создание удаленной ветки branch\_3

```
C:\lab3git\Lab_1.3>git fetch --all
From https://github.com/meiokava/Lab_1.3
* [new branch]      branch_3    -> origin/branch_3

C:\lab3git\Lab_1.3>
```

Рисунок 7.2 – Загрузка содержимого из удаленного репозитория перед отслеживанием

```
C:\lab3git\Lab_1.3>git checkout --track origin/branch_3
Switched to a new branch 'branch_3'
branch 'branch_3' set up to track 'origin/branch_3'.

C:\lab3git\Lab_1.3>_
```

Рисунок 7.3 – Создание ветки для отслеживания branch\_3

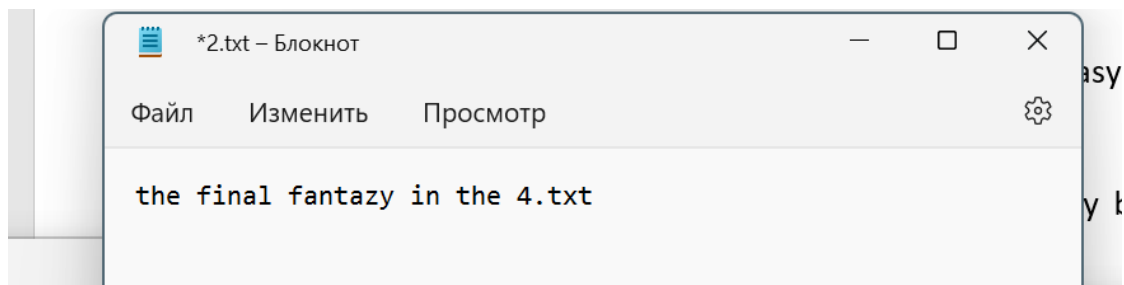


Рисунок 7.4 – Добавление новой строки в файл 2.txt

8. Было выполнено перемещение ветки main на ветку branch\_2, также изменения были отправлены изменения веток main и branch на GitHub

```

c:\lab3git\Lab_1.3>git checkout branch_2
switched to branch 'branch_2'

c:\lab3git\Lab_1.3>git rebase main
Current branch branch_2 is up to date.

c:\lab3git\Lab_1.3>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:\lab3git\Lab_1.3>git merge branch_2
Updating d8b85fb..258c23f
Fast-forward
 1.txt | 2 +-
 3.txt | 1 +
 2 files changed, 2 insertions(+), 1 deletion(-)

c:\lab3git\Lab_1.3>

```

Рисунок 8.1 – Результат после выполнения команд

```

c:\lab3git\Lab_1.3>git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 416 bytes | 416.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/meiokava/Lab_1.3.git
 5e56d40..258c23f  main -> main

c:\lab3git\Lab_1.3>git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.

c:\lab3git\Lab_1.3>git push
Everything up-to-date

c:\lab3git\Lab_1.3>

```

Рисунок 8.2 – Пуш веток

**Вывод:** были исследованы базовые возможности по работе с локальными и удаленными ветками GIT.

## Контрольные вопросы

### 1. Что такое ветка?

В Git ветки — это элемент повседневного процесса разработки. По сути ветки в Git представляют собой указатель на снимок изменений.

### 2. Что такое Head?

HEAD — указатель на текущий коммит или на текущую ветку (то есть, в любом случае, на коммит). Указывает на родителя коммита, который будет создан следующим

### **3. Способы создание веток**

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

### **4. Как узнать текущую ветку?**

При помощи команды `git branch -a`

### **5. Как переключаться между ветками?**

При помощи команды `git checkout <Название ветки>`

### **6. Что такое удаленная ветка?**

Удалённые ветки — это ссылки на состояние веток в ваших удалённых репозиториях. Это локальные ветки, которые нельзя перемещать. Они двигаются автоматически всякий раз, когда вы осуществляете связь по сети. Удалённые ветки действуют как закладки для напоминания о том, где ветки в удалённых репозиториях находились во время последнего подключения к ним.

### **7. Что такое ветка отслеживания?**

Проверка локальной ветви из удаленной ветви автоматически создает так называемую ветвь отслеживания. Ветви отслеживания-это локальные ветви, имеющие прямое отношение к удаленной ветви. Если вы находитесь в ветке отслеживания и набираете `git push`, Git автоматически знает, на какой сервер и ветку нажать.

### **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>`.

#### **12. Какие основные типы веток присутствуют в модели `git-flow`? Как организована работа с ветками в модели `git-flow`? В чем недостатки `git-flow`?**

В целом `GitFlow` состоит из двух постоянных веток и нескольких типов временных веток (В контексте `GitHub Flow`, `master` ветка — постоянная, а другие — временные).

В соответствии с этой моделью разработчики создают функциональную ветку и откладывают ее слияние с главной магистральной веткой до завершения работы над функцией. Такие долгосрочные функциональные ветки требуют тесного взаимодействия разработчиков при слиянии и создают повышенный риск отклонения от магистральной ветки.

Первая проблема: авторам приходится использовать ветку `develop` вместо `master`, поскольку `master` зарезервирован для кода, который отправляется в продакшен.

Вторая проблема процесса `git flow` – сложности, возникающие из-за веток для патчей и для релиза. Подобная структура может подойти некоторым организациям, но для абсолютного большинства она просто убийственно излишня.