

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО РАБОТЕ №3
дисциплины «Основы кроссплатформенного программирования»

Выполнил:
Кондратенко Даниил Витальевич
1 курс, группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. тех. наук, доцент,
доцент кафедры инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: основы ветвления Git.

Цель работы: исследовать базовые возможности по работе с локальными и удаленными ветками Git.

Порядок выполнения работы:

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

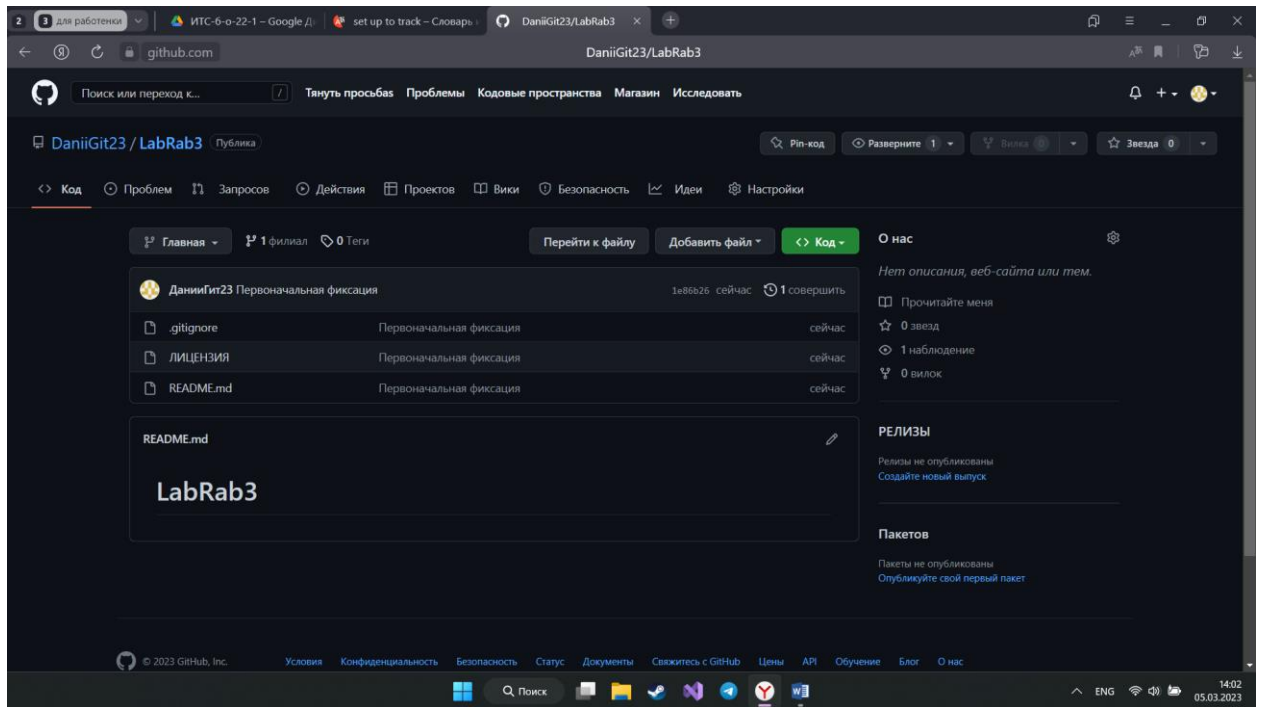
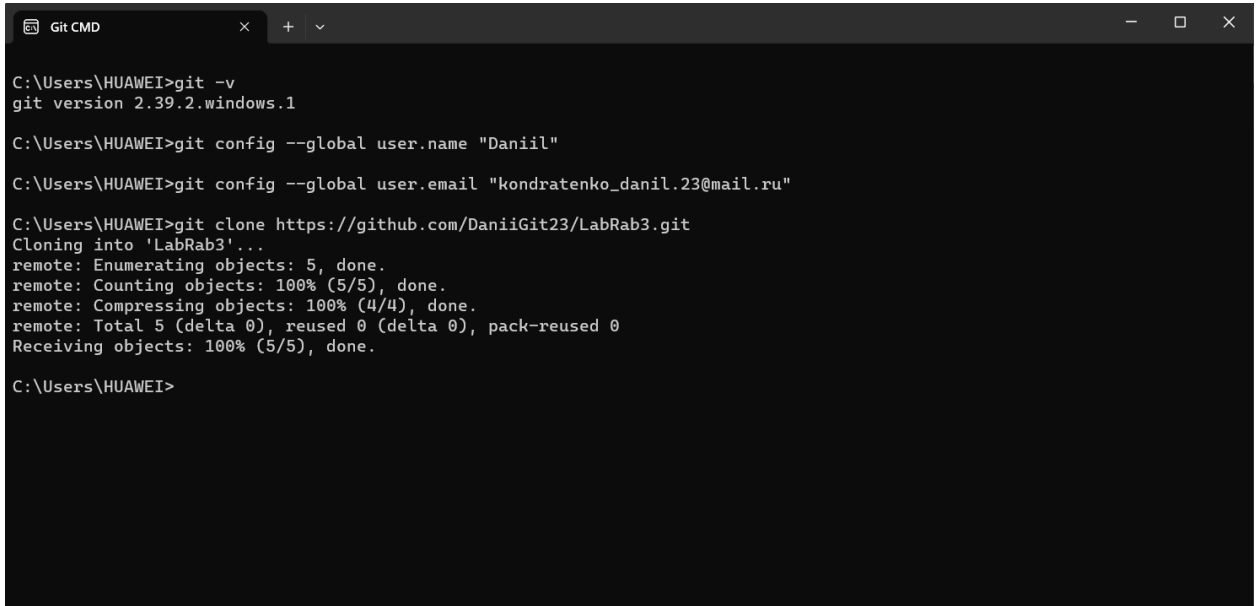


Рис. 1. Новый репозиторий.

2. Проклонировал репозиторий на свой компьютер.



```
C:\Users\HUAWEI>git -v
git version 2.39.2.windows.1

C:\Users\HUAWEI>git config --global user.name "Daniil"

C:\Users\HUAWEI>git config --global user.email "kondratenko_danil.23@mail.ru"

C:\Users\HUAWEI>git clone https://github.com/DaniiGit23/LabRab3.git
Cloning into 'LabRab3'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

C:\Users\HUAWEI>
```

Рис. 2. Клонирование.

3. Добавил 3 новых текстовых файла.

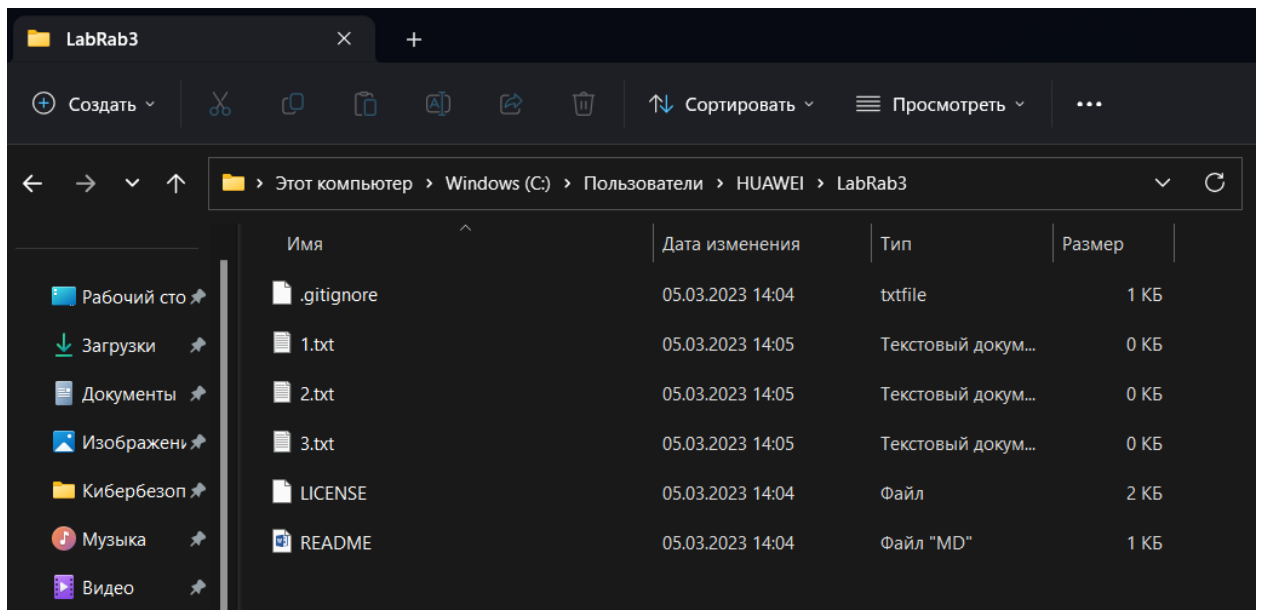


Рис. 3. Новые текстовые файлы.

4. Проиндексировал первый файл и сделать коммит с комментарием "add 1.txt file".

```
C:\Users\HUAWEI\LabRab3>git add 1.txt

C:\Users\HUAWEI\LabRab3>git commit -m "add 1.txt file"
[main dde39e8] add 1.txt file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt

C:\Users\HUAWEI\LabRab3>
```

Рис. 4. Добавление изменений и их фиксация.

5. Проиндексировал второй и третий файлы (добавил изменения в индекс).

```
Git CMD
C:\Users\HUAWEI\LabRab3>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   2.txt
        new file:   3.txt
```

Рис. 5. Добавление изменений.

6. Перезаписал уже сделанный коммит с новым комментарием "add 2.txt and 3.txt."

```
C:\Users\HUAWEI\LabRab3>git commit --amend -m "add 2.txt and 3.txt"
[main 6effaaf] add 2.txt and 3.txt
Date: Sun Mar 5 14:16:30 2023 +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\HUAWEI\LabRab3>
```

Рис. 6. Редактирование существующего коммита.

7. Создал новую ветку my_first_branch.

```
C:\Users\HUAWEI\LabRab3>git branch my_first_branch
```

Рис. 7. Новая ветка.

8. Перейти на ветку и создать новый файл in_branch.txt, закоммитить изменения.

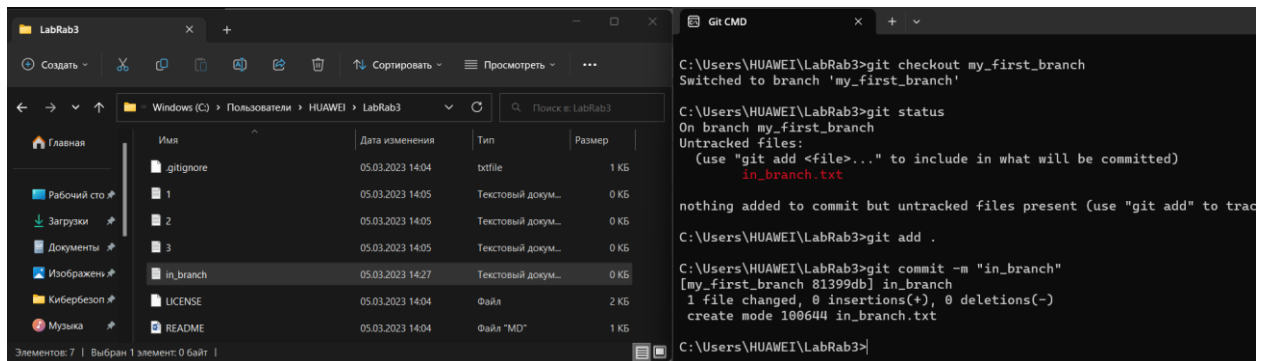


Рис. 8. Создание файла и фиксация.

9. Перешел вновь на новую ветку main.

```
C:\Users\HUAWEI\LabRab3>git checkout main
Switched to branch 'main'
```

Рис. 9. Переход на главную ветвь.

10. Создал и сразу перешел на ветку new_branch.

```
C:\Users\HUAWEI\LabRab3>git checkout -b new_branch
Switched to a new branch 'new_branch'
```

Рис. 10. Создание и сразу переход на ветку.

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

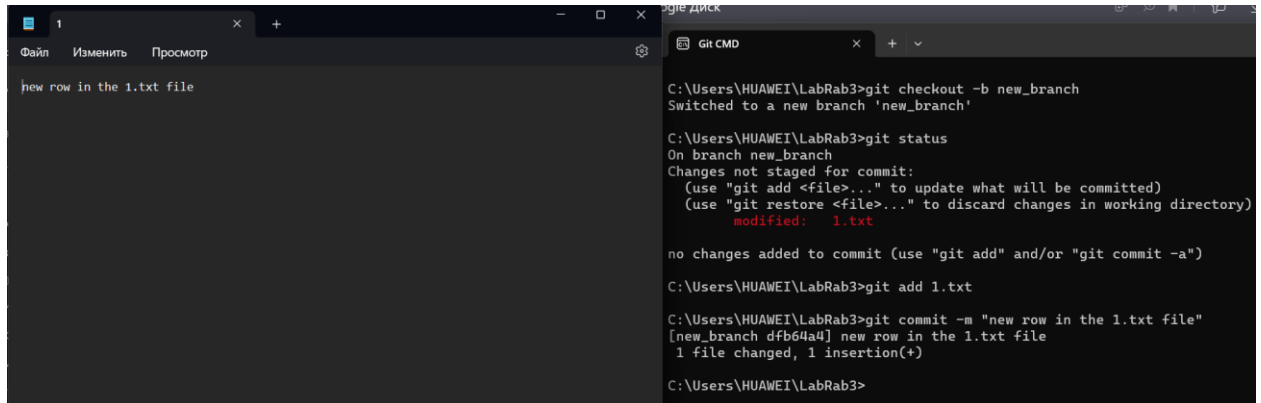


Рис. 11. Изменение в 1.txt фиксация этих изменений.

12. Перешел на ветку main и слил ветки main и my_first_branch, после чего слил ветки main и new_branch.

```
C:\Users\HUAWEI\LabRab3>git merge my_first_branch
Updating 6effaaf..81399db
Fast-forward
 in_branch.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 in_branch.txt

C:\Users\HUAWEI\LabRab3>git branch
* main
  my_first_branch
  new_branch

C:\Users\HUAWEI\LabRab3>git merge new_branch
Merge made by the 'ort' strategy.
 1.txt | 1 +
 1 file changed, 1 insertion(+)

C:\Users\HUAWEI\LabRab3>
```

Рис. 12. Слияние веток.

13. Удалил ветки my_first_branch и new_branch.

```
C:\Users\HUAWEI\LabRab3>git branch -d my_first_branch
Deleted branch my_first_branch (was 81399db).

C:\Users\HUAWEI\LabRab3>git branch -d new_branch
Deleted branch new_branch (was dfb64a4).
```

Рис. 13. Удаление веток.

14. Создал ветки branch_1 и branch_2.

```
C:\Users\HUAWEI\LabRab3>git branch branch_1

C:\Users\HUAWEI\LabRab3>git branch branch_2

C:\Users\HUAWEI\LabRab3>git branch
  branch_1
  branch_2
* main

C:\Users\HUAWEI\LabRab3>git checkout brange_1
error: pathspec 'brange_1' did not match any file(s) known to git
```

Рис. 14. Создание новых веток.

15. Перешел на ветку branch_1 и изменил файл 1.txt, удалил все содержимое и добавил текст “fix in the 1.txt”, изменил файл 3.txt, удалил все содержимое и добавил текст “fix in the 3.txt”, закоммитил изменения.

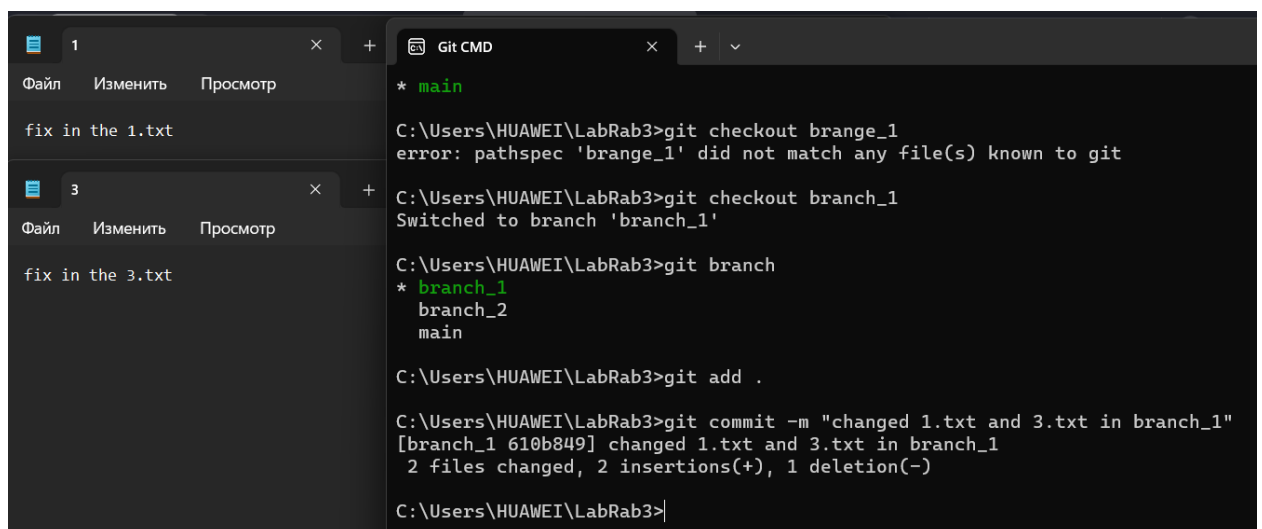


Рис. 15. Изменения в первой ветке и во второй ветка и фиксация изменений.

16. Перешел на ветку branch_2 и также изменил файл 1.txt, удалил все содержимое и добавил текст “My fix in the 1.txt”, изменил файл 3.txt, удалил все содержимое и добавил текст “My fix in the 3.txt”, закоммитил изменения.

```
C:\Users\HUAWEI\LabRab3>git checkout branch_2
Switched to branch 'branch_2'
```

Рис. 16. Переход на вторую ветку.

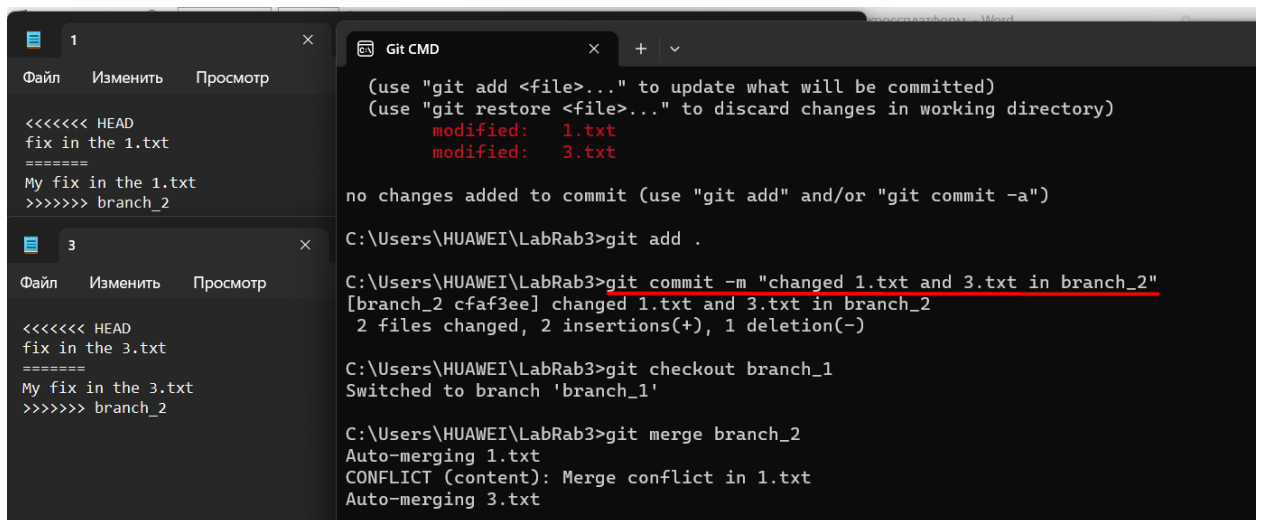


Рис. 17. Изменения файлов и их фиксация.

17. Слил изменения ветки branch_2 в ветку branch_1.

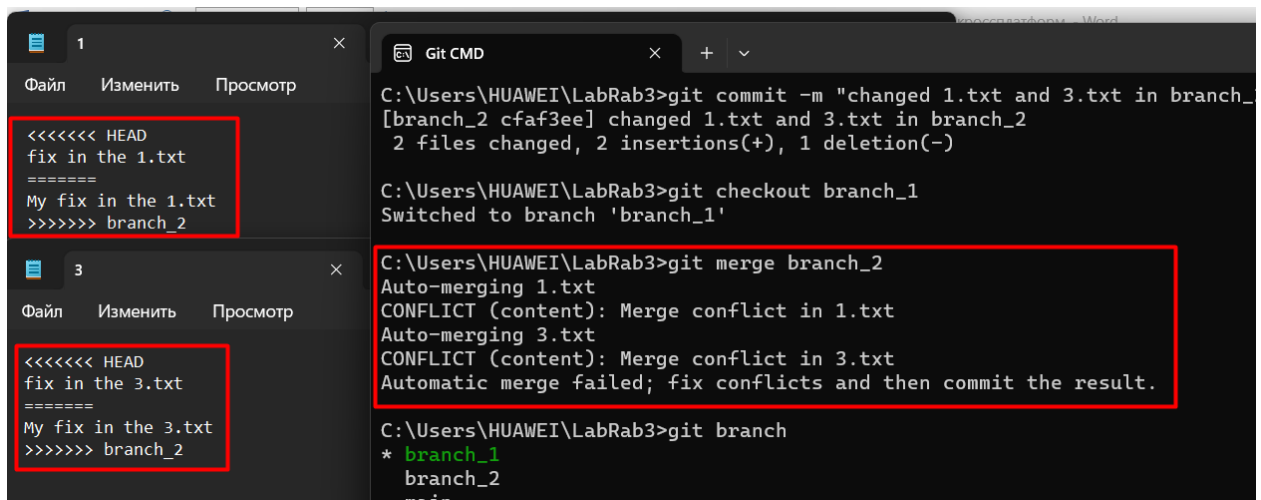


Рис. 18. Слияние веток.

18. Решил конфликт файла 1.txt в ручном режиме, а конфликт 3.txt используя команду `git mergetool` с помощью одной из доступных утилит, например `vimdiff`.

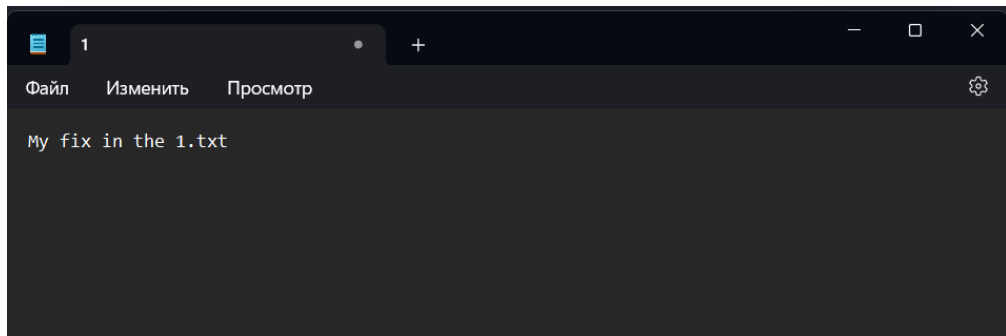


Рис. 19. Решение конфликта в ручную.

Необходимо изменить еще 3 текстовый файл, но уже с помощью инструмента.

Изменяю с помощью инструмента `vimdiff`.

```
C:\Users\HUAWEI\LabRab3>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:\Users\HUAWEI\LabRab3>git mergetool
Merging:
3.txt

Normal merge conflict for '3.txt':
  {local}: modified file
  {remote}: modified file
4 files to edit
```

Рис. 20. Решение конфликта с помощью инструмента.

19. Отправил branch_1 на удаленный сервер с помощью команды: *git push origin branch_1*.

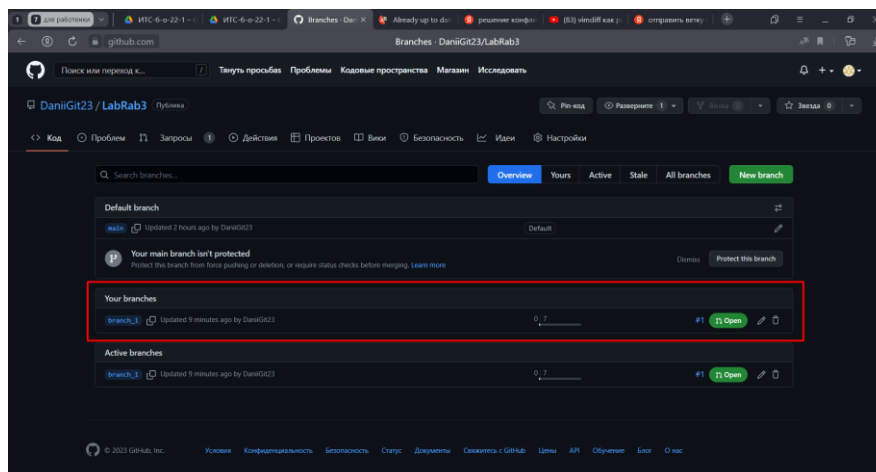


Рис. 21. Отправление ветки на удаленный сервер.

20. Создал средствами GitHub удаленную ветку branch_3.

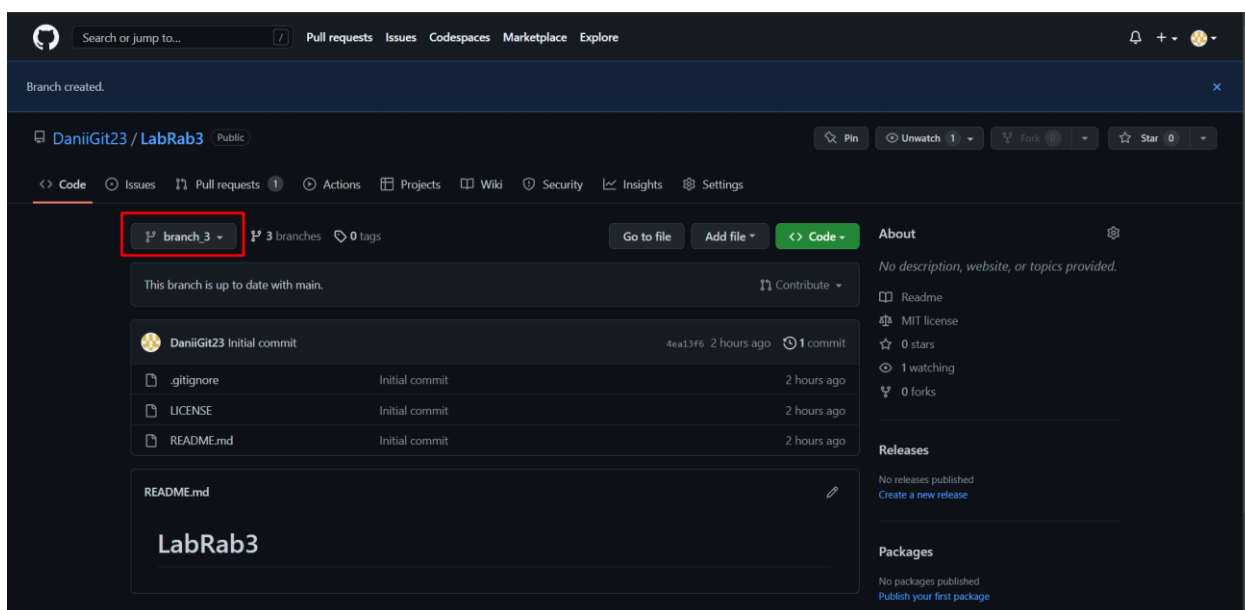


Рис. 22. Создание ветки на GitHub.

21. Создал в локальном репозитории ветку отслеживания удаленной ветки branch_3.

```
C:\Users\HUAWEI\LabRab3>git fetch origin
From https://github.com/DaniiGit23/LabRab3
* [new branch]      branch_3 -> origin/branch_3

C:\Users\HUAWEI\LabRab3>git checkout -b branch_3 origin/branch_3
Switched to a new branch 'branch_3'
branch 'branch_3' set up to track 'origin/branch_3'.

C:\Users\HUAWEI\LabRab3>|
```

Рис. 23. Создание ветки отслеживания.

22. Перешел на ветку branch_3 и добавил файл файл 2.txt строку "the final fantasy in the 4.txt file".

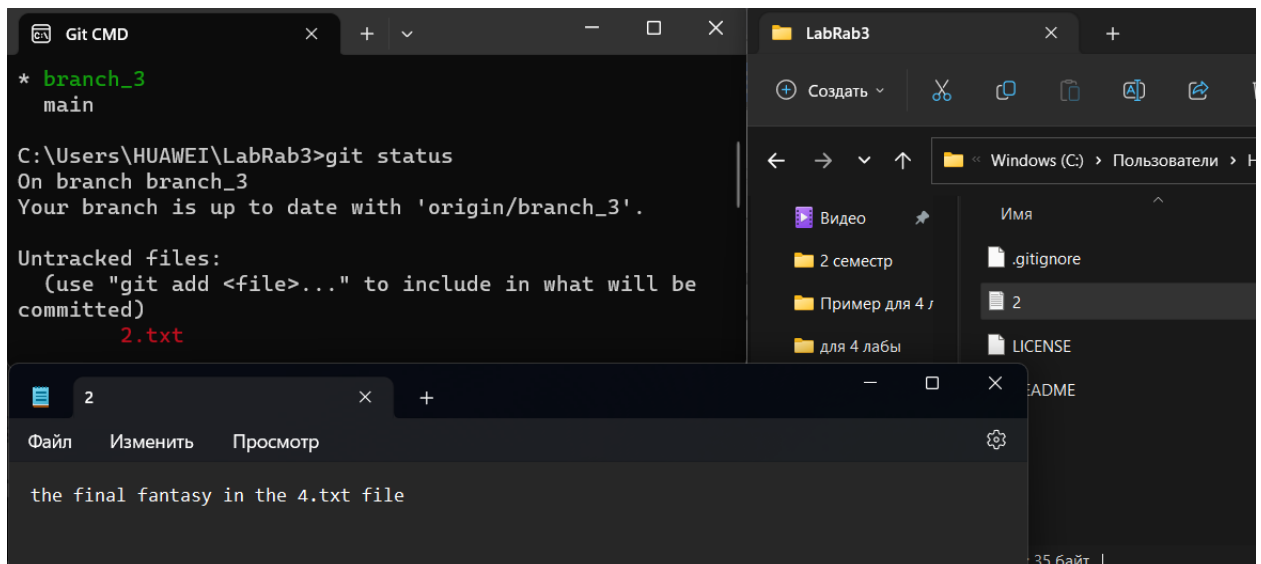


Рис. 24. Изменение файла во второй ветке.

23. Выполнил перемещение ветки main на ветку branch_2.

```
C:\Users\HUAWEI\LabRab3>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\HUAWEI\LabRab3>git rebase branch_2
Successfully rebased and updated refs/heads/main.

C:\Users\HUAWEI\LabRab3>git checkout branch_2
Switched to branch 'branch_2'

C:\Users\HUAWEI\LabRab3>git merge main
Already up to date.
```

Рис. 25. Перемещение веток.

24. Отправил изменения веток master и branch_2 на GitHub.

```
C:\Users\HUAWEI\LabRab3>git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 5 commits.
  (use "git push" to publish your local commits)

C:\Users\HUAWEI\LabRab3>git push origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/DaniiGit23/LabRab3.git
  4ea13f6..f401e73  main -> main

C:\Users\HUAWEI\LabRab3>git checkout branch_2
Switched to branch 'branch_2'

C:\Users\HUAWEI\LabRab3>git push 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/DaniiGit23/LabRab3/pull/new/branch_2
remote:
To https://github.com/DaniiGit23/LabRab3.git
 * [new branch]      branch_2 -> branch_2

C:\Users\HUAWEI\LabRab3>
```

Рис. 25. Отправка изменения веток на GitHub.

Ссылка на репозиторий: <https://github.com/DaniiGit23/LabRab3.git>

Ответы на контрольные вопросы:

1. Что такое ветка?

Почти каждая система контроля версий (СКВ) в какой-то форме поддерживает ветвление. Используя ветвление, Вы отклоняетесь от основной линии разработки и продолжаете работу независимо от неё, не вмешиваясь в основную линию.

2. Что такое HEAD?

HEAD в Git-это указатель на текущую ссылку ветви, которая, в свою очередь, является указателем на последний сделанный вами коммит или последний коммит, который был извлечен из вашего рабочего каталога.

3. Способы создания веток.

Создать ветку можно с помощью двух команд. Команда, которая просто создает ветку: `git branch "name_branch"`.

Команда, которая создает ветку и сразу же к ней переходит: `git checkout -b "name_branch"`.

4. Как узнать текущую ветку?

Текущую ветку можно узнать с помощью команды: `git branch`.

5. Как переключаться между ветками?

Между ветками можно переключаться с помощью команды: `git checkout "name_branch"`.

6. Что такое удаленная ветка?

Удаленные ветки - это ссылки на определённое состояние удалённых веток. Это локальные ветки, которые нельзя перемещать.

7. Что такое ветка отслеживания?

Отслеживаемые ветки — это локальные ветки, которые напрямую связаны с удалённой веткой.

Если, находясь на отслеживаемой ветке, вы наберёте `git push`, Git уже будет знать, на какой сервер и в какую ветку отправлять изменения.

8. Как создать ветку отслеживания?

Ветку отслеживания можно создать с помощью команды: `git checkout --track origin/<name_branch>`.

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

Отправить изменения из локальной ветки в удаленную можно с помощью команды: `git push origin <name_branch>`.

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

Команда `git fetch` получает с сервера все изменения, которых у вас ещё нет, но не будет изменять состояние вашей рабочей директории. Эта команда просто получает данные и позволяет вам самостоятельно сделать слияние. Тем не менее, существует команда `git pull`, которая в большинстве случаев является командой `git fetch`, за которой непосредственно следует команда `git merge`.

11. Как удалить локальную и удаленную ветки?

Для удаление локальной ветки используется команда: `git branch -d <name_branch>`

Для удаления удаленной ветки используется команда: `git push --delete origin/<name_branch>`

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

Существуют следующие типы ветвей:

- 1) ветви функциональностей;
- 2) ветви релизов;
- 3) ветви исправлений.

Ветви функциональностей (feature branches), также называемые иногда тематическими ветвями (topic branches), используются для разработки новых функций, которые должны появиться в текущем или будущем релизах.

Ветви релизов (release branches) используются для подготовки к выпуску новых версий продукта. Они позволяют расставить финальные точки над *i* перед выпуском новой версии.

Ветви для исправлений (hotfix branches) весьма похожи на ветви релизов (release branches), так как они тоже используются для подготовки новых выпусков продукта, разве лишь незапланированных.

Недостатки git flow: авторам приходится использовать ветку develop вместо master, поскольку master зарезервирован для кода.

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.