

**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 кроссплатформенного программирования»

Тема: «Основы ветвления Git»

Выполнил: студент 1 курса

группы ИВТ-б-о-21-1

Богдашов Артём

Ставрополь 2022

1. Создал репозиторий лицензии MIT. После чего клонировал его на ПК:

The screenshot shows the GitHub 'Create a new repository' page. At the top, it says 'Create a new repository' and explains that a repository contains all project files, including the revision history. It also provides a link to 'Import a repository'. Below this, there are two main input fields: 'Owner' and 'Repository name'. The 'Owner' field is set to 'Artem0622' and the 'Repository name' field is set to 'Lab1.3', which has a green checkmark indicating it's valid. A note below these fields says 'Great repository names are short and memorable. Need inspiration? How about curly-adventure?'. There is a 'Description (optional)' text area below. Further down, there are two radio button options for repository visibility: 'Public' (selected) and 'Private'. The 'Public' option is described as 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option is described as 'You choose who can see and commit to this repository.' Below these options, there is a section 'Initialize this repository with:' which includes a checkbox for 'Add a README file' (unchecked) and a section for 'Add .gitignore' with a dropdown menu set to '.gitignore template: None'. There is also a 'Choose a license' section with a dropdown menu set to 'License: MIT License'. At the bottom, there is a note 'You are creating a public repository in your personal account.' and a green 'Create repository' button.

Рисунок 1.1 Создание репозитория

```
Git CMD
C:\Users\Kw1\>cd/d C:\Users\Kw1\Desktop\1.3
C:\Users\Kw1\Desktop\1.3>git clone https://github.com/Artem0622/Lab1.3.git
Cloning into 'Lab1.3'...
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\Kw1\Desktop\1.3>
```

Рисунок 1.2 Клонирование репозитория

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”:

Компьютер > Рабочий стол > 1.3 > Lab1.3

Имя	Дата изменения	Тип	Размер
.git	22.05.2022 20:59	Папка с файлами	
1.txt	22.05.2022 21:08	Текстовый докум...	0 КБ
2.txt	22.05.2022 21:09	Текстовый докум...	0 КБ
3.txt	22.05.2022 21:09	Текстовый докум...	0 КБ
LICENSE	22.05.2022 20:59	Файл	2 КБ

Рисунок 2.1 Созданные файлы в репозитории

```

Git CMD
C:\Users\Kwil\Desktop\1.3\Lab1.3>cd/d C:\Users\Kwil\Desktop\1.3\Lab1.3
C:\Users\Kwil\Desktop\1.3\Lab1.3>git add 1.txt
C:\Users\Kwil\Desktop\1.3\Lab1.3>git commit -m "add 1.txt file"
[main 5d031b5] add 1.txt file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt
C:\Users\Kwil\Desktop\1.3\Lab1.3>_

```

Рисунок 2.2 1-й коммит

```

C:\Users\Kwil\Desktop\1.3\Lab1.3>git add .
C:\Users\Kwil\Desktop\1.3\Lab1.3>git commit --amend -m "add 2.txt and 3.txt"
[main c465028] add 2.txt and 3.txt
Date: Sun May 22 21:34:43 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\Kwil\Desktop\1.3\Lab1.3>_

```

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`, закоммитил изменения. Вернулся на ветку `master`, создал и сразу перешел на ветку `new_branch`, сделал изменения в файле `1.txt` и закоммитил изменения.

```

Git CMD
create mode 100644 3.txt
C:\Users\Kwil\Desktop\1.3\Lab1.3>git branch my_first_branch
C:\Users\Kwil\Desktop\1.3\Lab1.3>git checkout my_first_branch
Switched to branch 'my_first_branch'
C:\Users\Kwil\Desktop\1.3\Lab1.3>git branch
main
* my_first_branch
C:\Users\Kwil\Desktop\1.3\Lab1.3>

```

Рисунок 3.1 Создание и переход на новую ветку

Компьютер > Рабочий стол > 1.3 > Lab1.3

Имя	Дата изменения	Тип	Размер
.git	22.05.2022 21:44	Папка с файлами	
1.txt	22.05.2022 21:32	Текстовый докум...	0 КБ
2.txt	22.05.2022 21:32	Текстовый докум...	0 КБ
3.txt	22.05.2022 21:33	Текстовый докум...	0 КБ
in_branch.txt	22.05.2022 21:49	Текстовый докум...	0 КБ
LICENSE	22.05.2022 21:32	Файл	2 КБ

Рисунок 3.2 Добавление файла в репозиторий

```

Git CMD
C:\Users\Kw1\Desktop\1.3\Lab1.3>git add .
C:\Users\Kw1\Desktop\1.3\Lab1.3>git commit -m "add in_branch.txt"
[my_first_branch 091587c] add in_branch.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 in_branch.txt
C:\Users\Kw1\Desktop\1.3\Lab1.3>_

```

Рисунок 3.3 Коммит изменений

```

C:\Users\Kw1\Desktop\1.3\Lab1.3>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\Kw1\Desktop\1.3\Lab1.3>git checkout -b new_branch
"git" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\Kw1\Desktop\1.3\Lab1.3>git checkout -b new_branch
Switched to a new branch 'new_branch'

C:\Users\Kw1\Desktop\1.3\Lab1.3>git branch
main
my_first_branch
* new_branch
C:\Users\Kw1\Desktop\1.3\Lab1.3>_

```

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

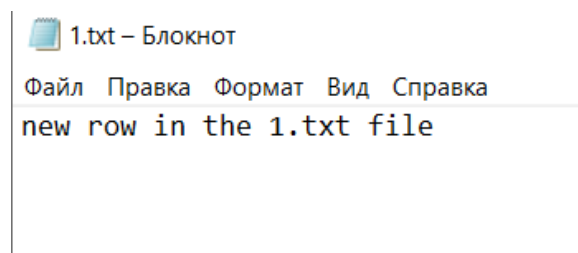


Рисунок 3.5 Изменения в файле 1.txt

```

Git CMD
* new_branch
C:\Users\Kw1\Desktop\1.3\Lab1.3>git commit -am "fix 1.txt"
[new_branch 4efc9b5] 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:

```
Git CMD
C:\Users\Kw1\Desktop\1.3\Lab1.3>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\Kw1\Desktop\1.3\Lab1.3>git merge my_first_branch
Updating c465028..091587c
Fast-forward
 in_branch.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 in_branch.txt

C:\Users\Kw1\Desktop\1.3\Lab1.3>git branch
* main
  my_first_branch
  new_branch

C:\Users\Kw1\Desktop\1.3\Lab1.3>git merge new_branch
Merge made by the 'ort' strategy.
 1.txt | 1 +
 1 file changed, 1 insertion(+)

C:\Users\Kw1\Desktop\1.3\Lab1.3>git branch -d my_first_branch
Deleted branch my_first_branch (was 091587c).

C:\Users\Kw1\Desktop\1.3\Lab1.3>git branch -d new_branch
Deleted branch new_branch (was 4efc9b5).

C:\Users\Kw1\Desktop\1.3\Lab1.3>
```

Рисунок 4.1 Слияние и удаления 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:

```
Git CMD
C:\Users\Kw1\Desktop\1.3\Lab1.3>git branch branch_1
C:\Users\Kw1\Desktop\1.3\Lab1.3>git branch branch_2
C:\Users\Kw1\Desktop\1.3\Lab1.3>git branch
  branch_1
  branch_2
* main
```

Рисунок 5.1 Создание веток branch_1 и branch_2

```
C:\Users\Kw1\Desktop\1.3\Lab1.3>git checkout branch_1
Switched to branch 'branch_1'
```

Рисунок 5.2 Переход на ветку branch_1

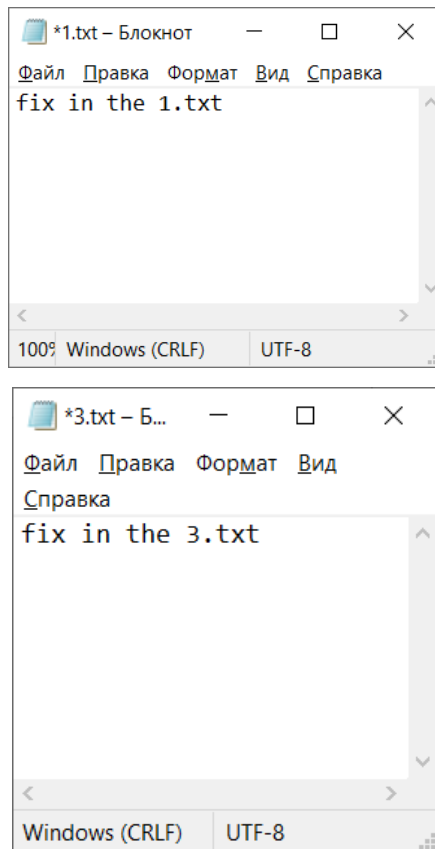


Рисунок 5.3 Изменения в файлах 1.txt и 3.txt

```
Git CMD
C:\Users\Kwi1\Desktop\1.3\Lab1.3>git add .
C:\Users\Kwi1\Desktop\1.3\Lab1.3>git commit -m "fix 1.txt & 3.txt in branch_1"
[branch_1 47deef9] fix 1.txt & 3.txt in branch_1
2 files changed, 2 insertions(+), 1 deletion(-)
```

Рисунок 5.4 Коммит изменений

```
Git CMD
C:\Users\Kwi1\Desktop\1.3\Lab1.3>git checkout branch_2
Switched to branch 'branch_2'
```

Рисунок 5.5 Переход на ветку branch_2

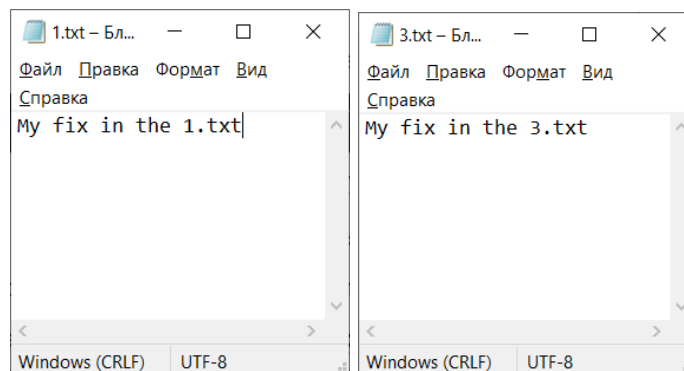


Рисунок 5.6 Изменения в файлах 1.txt и 3.txt

```
C:\Users\Kwi1\Desktop\1.3\Lab1.3>git commit -am "fix1.txt & 3.txt in branch_2"
[branch_2 749830b] fix1.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:

```
Git CMD

C:\Users\Kwi1\Desktop\1.3\Lab1.3>git checkout branch_1
Switched to branch 'branch_1'

C:\Users\Kwi1\Desktop\1.3\Lab1.3>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.
```

Рисунок 6.1 Слияние веток

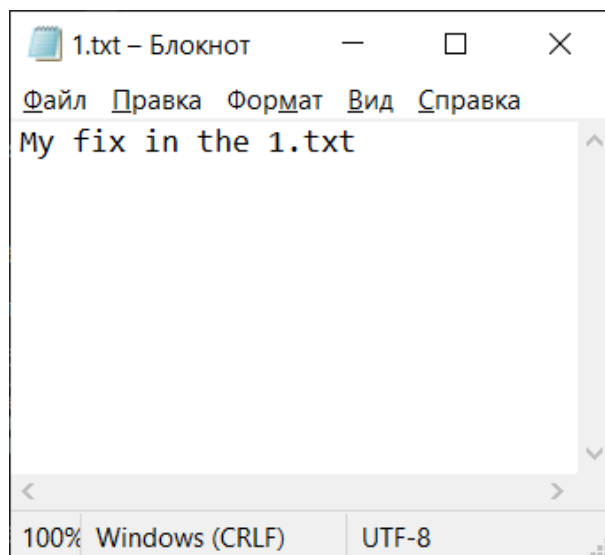


Рисунок 6.2 Решение конфликта в ручном режиме

```
Git CMD

C:\Users\Kwi1\Desktop\1.3\Lab1.3>git add 1.txt
C:\Users\Kwi1\Desktop\1.3\Lab1.3>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, так как конфликт решен

```
Git CMD - "C:\Program Files\Git\cmd\git.exe" mergetool

C:\Users\Kwil\Desktop\1.3\Lab1.3>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

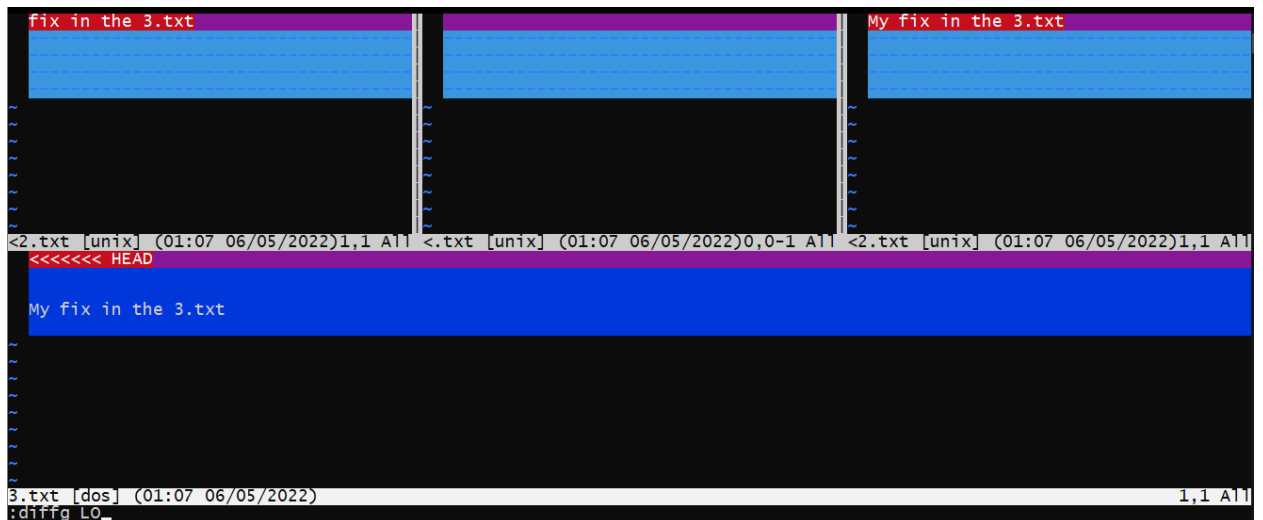


Рисунок 6.5 Команда :diffg LO

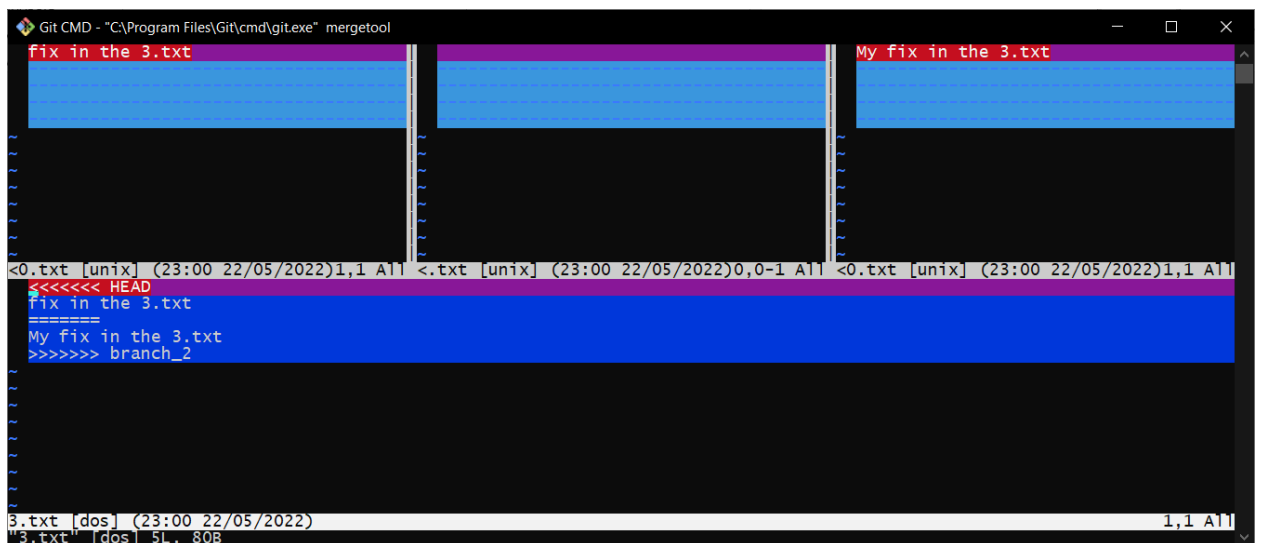


Рисунок 6.6 Команда :wga


```
Git CMD

C:\Users\Kwil>cd/d C:\Users\Kwil\Desktop\1.3\Lab1.3

C:\Users\Kwil\Desktop\1.3\Lab1.3>git push --set-upstream origin branch_1
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 12 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (14/14), 1.15 KiB | 1.15 MiB/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/Artem0622/Lab1.3/pull/new/branch_1
remote:
To https://github.com/Artem0622/Lab1.3.git
 * [new branch]      branch_1 -> branch_1
branch 'branch_1' set up to track 'origin/branch_1'.
```

Рисунок 6.7 Пуш ветки на уд. Репозиторий

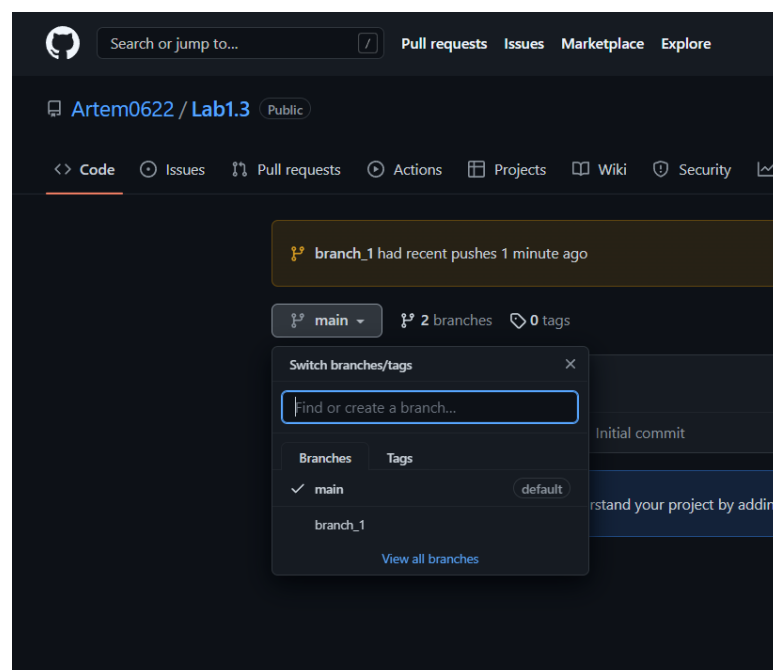


Рисунок 6.8 Изменения на уд. репозитории

7. Создал средствами Git удаленную ветку branch_4, создал в локальном 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”:

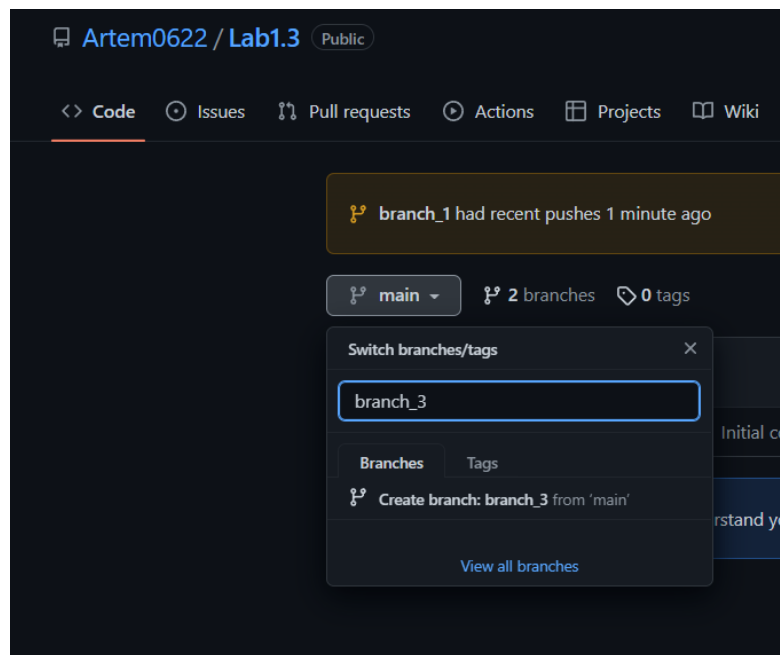


Рисунок 7.1 Создание уд. ветки branch_3

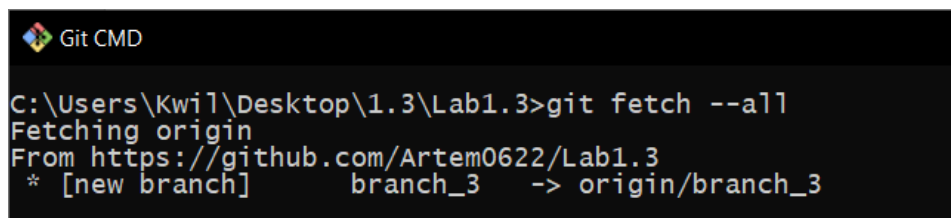


Рисунок 7.2 Загрузка содержимого из уд. репозитория перед отслеживанием

```
C:\Users\Kwi1\Desktop\1.3\lab\Lab1.3>git checkout --track origin/branch_3
Switched to a new branch 'branch_3'
M   1.txt
M   3.txt
branch 'branch_3' set up to track 'origin/branch_3'.
```

Рисунок 7.3 Создание ветки отслеживания branch_3

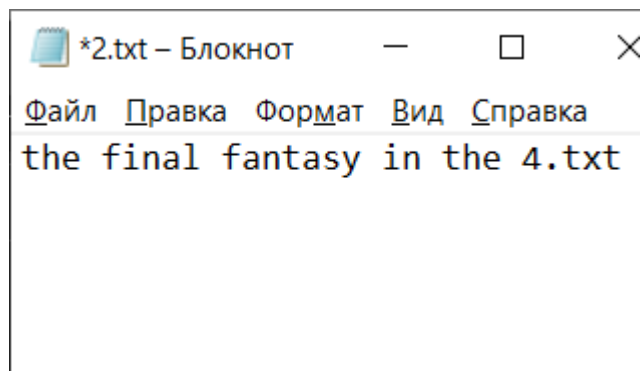


Рисунок 7.4 Добавление в 2.txt строку “the final fantasy in the 4.txt file” на ветке branch_3

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

```
Git CMD
C:\Users\Kwil\Desktop\1.3\Lab1.3>git checkout branch_2
Switched to branch 'branch_2'

C:\Users\Kwil\Desktop\1.3\Lab1.3>git rebase main
Current branch branch_2 is up to date.

C:\Users\Kwil\Desktop\1.3\Lab1.3>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\Kwil\Desktop\1.3\Lab1.3>git merge branch_2
Updating 2664b92..749830b
Fast-forward
 1.txt | 2 + -
 3.txt | 1 +
 2 files changed, 2 insertions(+), 1 deletion(-)

C:\Users\Kwil\Desktop\1.3\Lab1.3>
```

Рисунок 8.1 Результат выполнения команд

```
C:\Users\Kwil\Desktop\1.3\Lab1.3>git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 400 bytes | 400.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Artem0622/Lab1.3.git
 e4b7782..749830b main -> main
```

Рисунок 8.2 Пуш веток

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 удаленном 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 удаленная ветка?

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

7) Что такое ветка отслеживания?

Это ветка в локальном 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 удаления удаленной ветки существует команда `git push origin -d <название_ветки>`.

