

**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

Хашиев Адам Мухарбекович

Ставрополь 2022

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

Owner * Repository name *

AdamKh / rep_1.3 ✓

Great repository names are short and memorable. Need inspiration? How about **animated-train**?

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.](#)

.gitignore template: None ▾

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

License: MIT License ▾

i You are creating a public repository in your personal account.

Create repository

Рисунок 1.1 Создание репозитория

```
C:\Users\adamk\OneDrive\Рабочий стол>git clone https://github.com/AdamKh/rep_1.3.git
Cloning into 'rep_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.
```

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”:

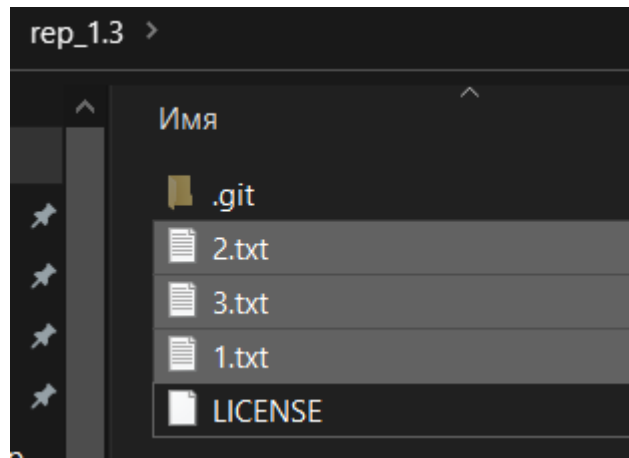


Рисунок 2.1 Созданные файлы в репозитории

```
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git add 1.txt
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git commit -m "add 1.txt file"
[main cb350fb] add 1.txt file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt
```

Рисунок 2.2 1-й коммит

```
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git add .
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git commit --amend -m "add 2.txt and 3.txt"
[main aea7127] add 2.txt and 3.txt
Date: Thu Apr 21 23:37:08 2022 +0300
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt
create mode 100644 2.txt
create mode 100644 3.txt
```

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. После этого вернулся на ветку master, создал и сразу перешел на ветку new_branch, сделал изменения в файле 1.txt и закоммитил изменения.

```
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git branch my_first_branch
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git checkout my_first_branch
Switched to branch 'my_first_branch'

C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git branch
  main
* my_first_branch
```

Рисунок 3.1 Создание и переход на новую ветку

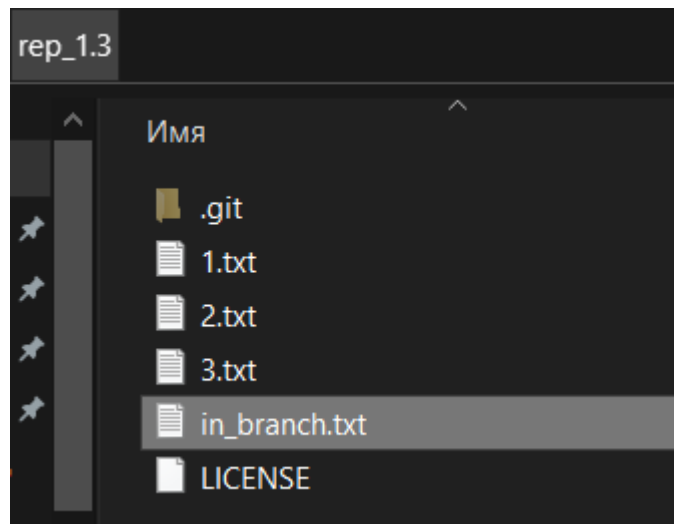


Рисунок 3.2 Добавление файла в репозиторий

```
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git add .
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git commit -m "add in_branch.txt"
[my_first_branch 941bfac] add in_branch.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 in_branch.txt
```

Рисунок 3.3 Коммит изменений

```
C:\Users\adamk\OneDrive\Рабочий стол\rep_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:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git checkout -b new_branch
Switched to a new branch 'new_branch'
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git branch
main
my_first_branch
* new_branch
```

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

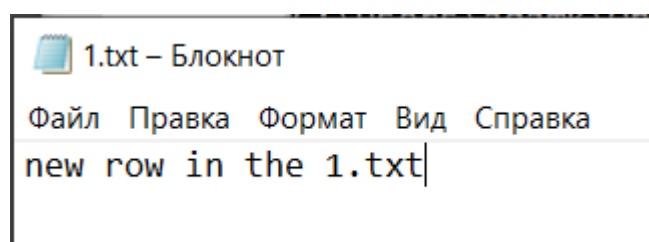


Рисунок 3.5 Изменения в файле 1.txt

```
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git commit -am "fix 1.txt"
[new_branch a261d74] 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\adamk\OneDrive\Рабочий стол\rep_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:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git merge my_first_branch
Updating aea7127..941bfac
Fast-forward
 in_branch.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 in_branch.txt

C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git branch
* main
  my_first_branch
  new_branch

C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git merge new_branch
Merge made by the 'ort' strategy.
 1.txt | 1 +
 1 file changed, 1 insertion(+)

C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git branch -d my_first_branch
Deleted branch my_first_branch (was 941bfac).

C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git branch -d new_branch
Deleted branch new_branch (was a261d74).
```

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:

```
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git branch branch_1
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git branch branch_2
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git branch
  branch_1
  branch_2
* main
```

Рисунок 5.1 Создание веток branch_1 и branch_2

```
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git checkout branch_1
Switched to branch 'branch_1'
```

Рисунок 5.2 Переход на ветку branch_1

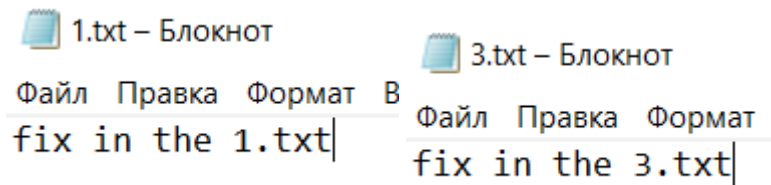


Рисунок 5.3 Изменения в файлах 1.txt и 3.txt

```
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git add .  
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git commit -m "fix 1.txt & 3.txt in branch_1"  
[branch_1 14561a1] fix 1.txt & 3.txt in branch_1  
2 files changed, 2 insertions(+), 1 deletion(-)
```

Рисунок 5.4 Коммит изменений

```
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git checkout branch_2  
Switched to branch 'branch_2'
```

Рисунок 5.5 Переход на ветку branch_2

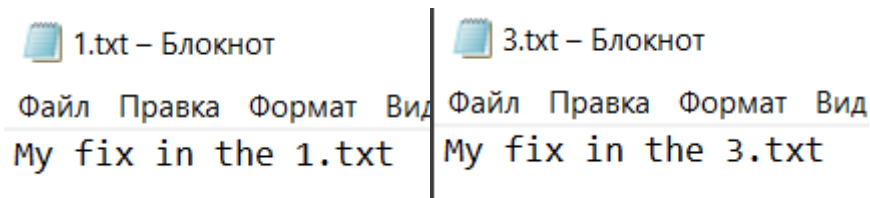


Рисунок 5.6 Изменения в файлах 1.txt и 3.txt

```
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git commit -am "fix 1.txt & 3.txt in branch_2"  
[branch_2 d20db22] 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\adamk\OneDrive\Рабочий стол\rep_1.3>git checkout branch_1  
Switched to branch 'branch_1'  
  
C:\Users\adamk\OneDrive\Рабочий стол\rep_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.
```

Рисунок 6.1 Слияние веток

1.txt – Блокнот

Файл Правка Формат В
My fix in the 1.txt

Рисунок 6.2 Решение конфликта в ручном режиме

```
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git add 1.txt
C:\Users\adamk\OneDrive\Рабочий стол\rep_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
```

Рисунок 6.3 Добавление 1.txt в индекс, так как конфликт решен

```
C:\Users\adamk\OneDrive\Рабочий стол\rep_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

The screenshot shows a three-pane diff tool interface. The top left pane displays 'fix in the 3.txt' on a blue background. The top right pane displays 'My fix in the 3.txt' on a blue background. The bottom pane shows the merged result, which is 'My fix in the 3.txt' on a blue background. The interface includes status bars at the bottom of each pane and a central status bar. The central status bar shows '<2.txt [unix] (01:07 06/05/2022)1,1 A11' and '<.txt [unix] (01:07 06/05/2022)0,0-1 A11'. The bottom status bar shows '3.txt [dos] (01:07 06/05/2022) 1,1 A11' and ':diffg LO_'. The interface also includes a 'HEAD' label and a 'diffg LO_' label.

Рисунок 6.5 Команда :diffg LO

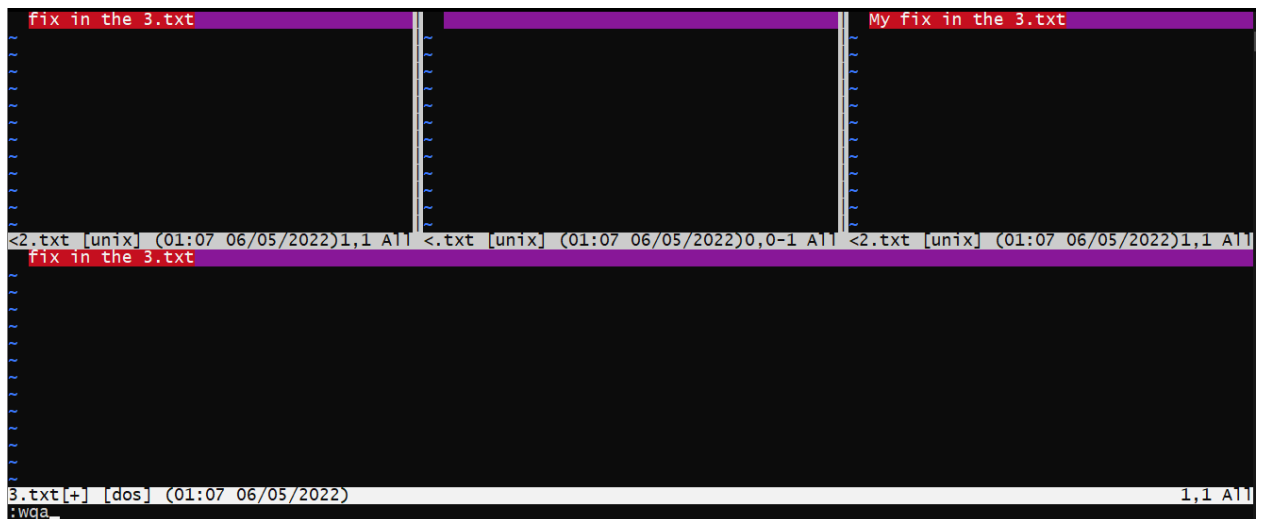


Рисунок 6.6 Команда :wga

```
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git push --set-upstream origin branch_1
info: please complete authentication in your browser...
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 8 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (21/21), 1.66 KiB | 568.00 KiB/s, done.
Total 21 (delta 7), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (7/7), done.
remote:
remote: Create a pull request for 'branch_1' on GitHub by visiting:
remote:   https://github.com/AdamKh/rep_1.3/pull/new/branch_1
remote:
To https://github.com/AdamKh/rep_1.3.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

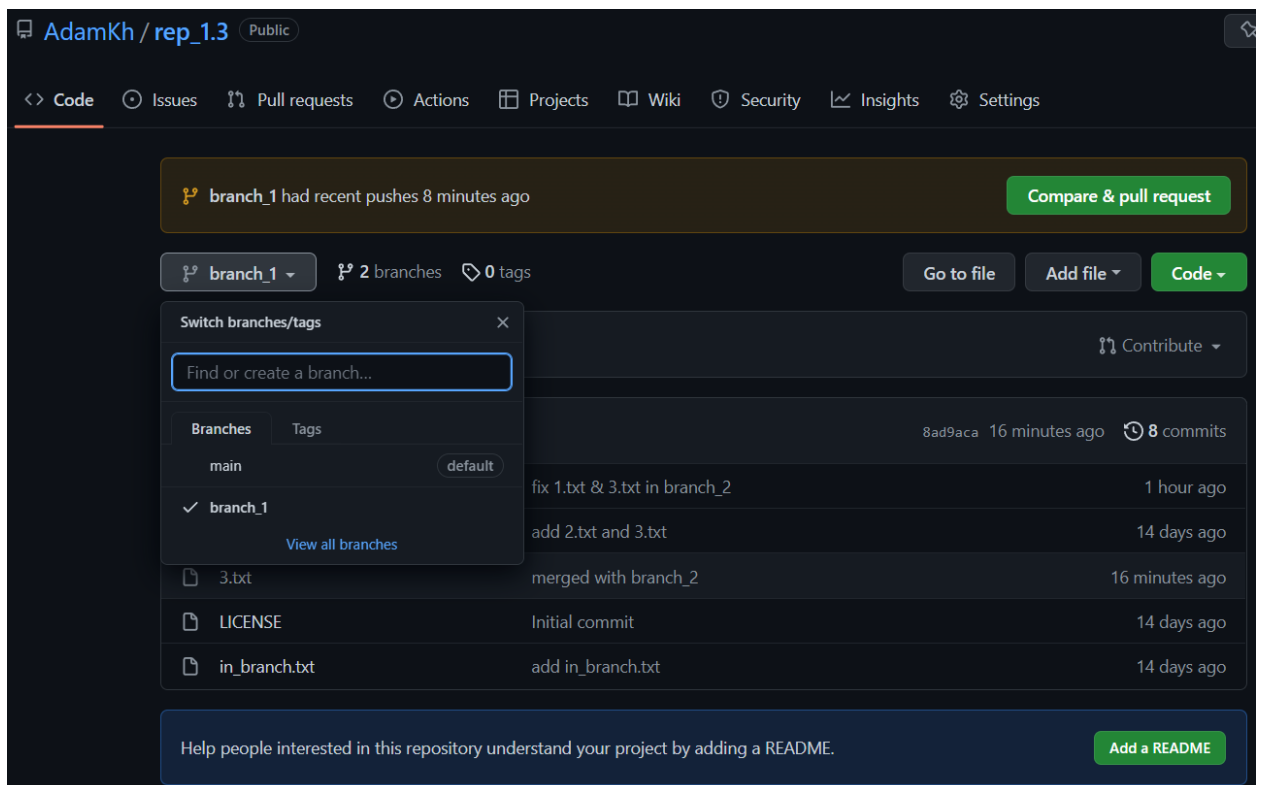


Рисунок 6.8 Изменения на уд. репозитории

7. Создал средствами 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”:

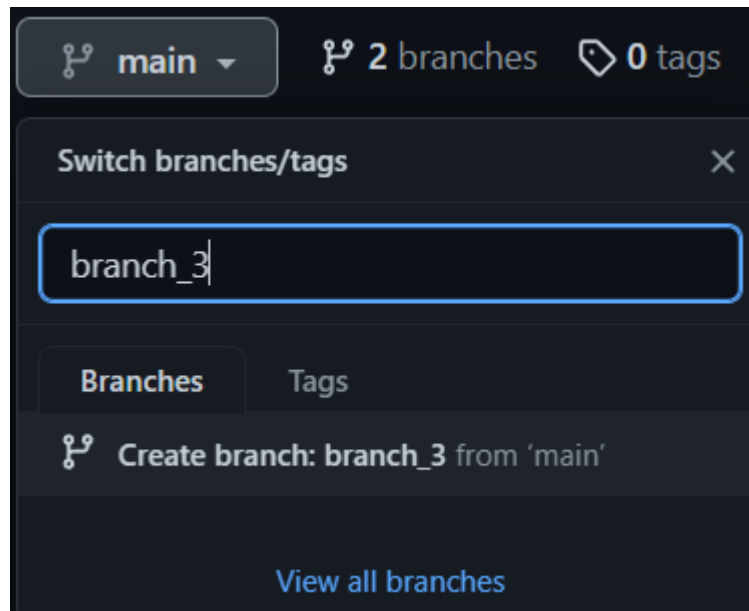


Рисунок 7.1 Создание уд. ветки branch_3

```
C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git fetch --all
Fetching origin
From https://github.com/AdamKh/rep_1.3
* [new branch]      branch_3    -> origin/branch_3
```

Рисунок 7.2 Загрузка содержимого из уд. репозитория перед отслеживанием

```
C:\Users\adamk\OneDrive\Рабочий стол\rep_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:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>
```

Рисунок 7.3 Создание ветки отслеживания branch_3

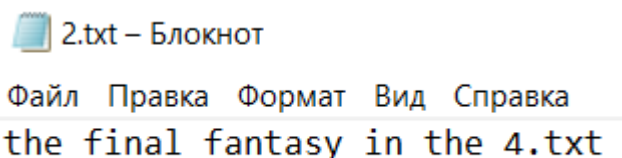


Рисунок 7.4 Добавление в 2.txt строку “the final fantasy in the 4.txt file” на ветке branch_3

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

```

C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git checkout branch_2
Switched to branch 'branch_2'

C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git rebase main
Current branch branch_2 is up to date.

C:\Users\adamk\OneDrive\Рабочий стол\rep_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:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git merge branch_2
Updating 77f29f4..ddb0601
Fast-forward
 1.txt | 2 +-
 3.txt | 1 +
 2 files changed, 2 insertions(+), 1 deletion(-)

```

Рисунок 8.1 Результат выполнения команд

```

C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/AdamKh/rep_1.3.git
 bc35384..ddb0601  main -> main

C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.

C:\Users\adamk\OneDrive\Рабочий стол\rep_1.3>git push
Everything up-to-date

```

Рисунок 8.2 Пуш веток

Ссылка на репозиторий Git Hub: https://github.com/AdamKh/rep_1.3

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 удаление удаленной ветки существует команда `git push origin -d <название_ветки>`.