

**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 2 курса  
группы ПИЖ-б-о-21-1  
Пуценко Иван Алексеевич

Ставрополь 2022

1. 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”:

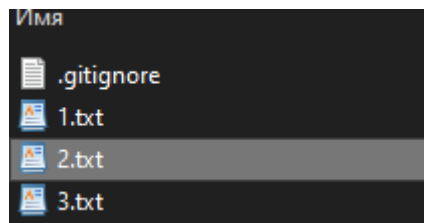


Рисунок 1.1 Созданные файлы в репозитории

```
C:\Users\FonK\Desktop\ОПИ>git clone https://github.com/kvakaet/labRab_3.git
Cloning into 'labRab_3'...
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\FonK\Desktop\ОПИ>cd labRab_3

C:\Users\FonK\Desktop\ОПИ\labRab_3>git add 1.txt

C:\Users\FonK\Desktop\ОПИ\labRab_3>git commit -m "add 1.txt file"
[main 63a71c8] add 1.txt file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt
```

Рисунок 1.2 1-й коммит

```
C:\Users\FonK\Desktop\ОПИ\labRab_3>git commit --amend -m "add 2.txt and 3.txt"
[main d063f0e] add 2.txt and 3.txt
Date: Tue Dec 6 02:11:03 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 1.3 Изменение 1-го коммита

2. 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\FonK\Desktop\ОПИ\labRab_3>git branch my_first_branch
```

```
C:\Users\FonK\Desktop\ОПИ\labRab_3>git checkout my_first_branch  
Switched to branch 'my_first_branch'
```

Рисунок 2.1 Создание и переход на новую ветку

```
C:\Users\FonK\Desktop\ОПИ\labRab_3>git add .

C:\Users\FonK\Desktop\ОПИ\labRab_3>git commit -m "add in_branch.txt"
[my_first_branch 3972cf9] add in_branch.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 in_branch.txt
```

Рисунок 2.2 Коммит изменений

```
C:\Users\FonK\Desktop\ОПИ\labRab_3>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\Users\FonK\Desktop\ОПИ\labRab_3>git checkout -b new_branch
Switched to a new branch 'new_branch'
```

Рисунок 2.3 Переход на ветку main и создание + переход на новую ветку

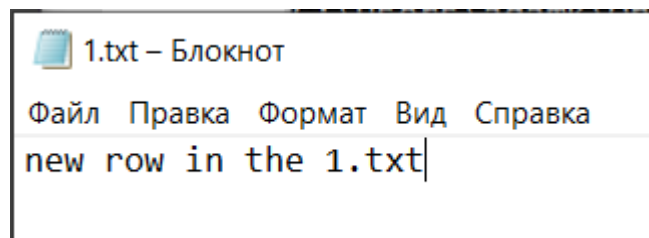


Рисунок 2.4 Изменения в файле 1.txt

```
C:\Users\FonK\Desktop\ОПИ\labRab_3>git branch
main
my_first_branch
* new_branch

C:\Users\FonK\Desktop\ОПИ\labRab_3>git add .

C:\Users\FonK\Desktop\ОПИ\labRab_3>git commit -m "fix 1.txt"
[new_branch 072a914] fix 1.txt
1 file changed, 1 insertion(+)
```

Рисунок 2.5 Коммит изменений

3. 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\FonK\Desktop\ОПИ\labRab_3>git checkout main
Switched to branch 'main'
M       README.md
Your branch is up to date with 'origin/main'.

C:\Users\FonK\Desktop\ОПИ\labRab_3>git merge my_first_branch
Updating d063f0e..3972cf9
Fast-forward
 in_branch.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 in_branch.txt

C:\Users\FonK\Desktop\ОПИ\labRab_3>git merge new_branch
Merge made by the 'ort' strategy.
 1.txt | 1 +
 1 file changed, 1 insertion(+)

C:\Users\FonK\Desktop\ОПИ\labRab_3>git branch -d my_first_branch
Deleted branch my_first_branch (was 3972cf9).

C:\Users\FonK\Desktop\ОПИ\labRab_3>git branch -d new_branch
Deleted branch new_branch (was 072a914).

C:\Users\FonK\Desktop\ОПИ\labRab_3>git branch
* main
```

Рисунок 3.1 Слияние и удаления веток

4. 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\FonK\Desktop\ОПИ\labRab_3>git branch branch_1
C:\Users\FonK\Desktop\ОПИ\labRab_3>git branch branch_2
```

Рисунок 4.1 Создание веток branch\_1 и branch\_2

```
C:\Users\FonK\Desktop\ОПИ\labRab_3>git checkout branch_1
Switched to branch 'branch_1'
```

Рисунок 4.2 Переход на ветку branch\_1



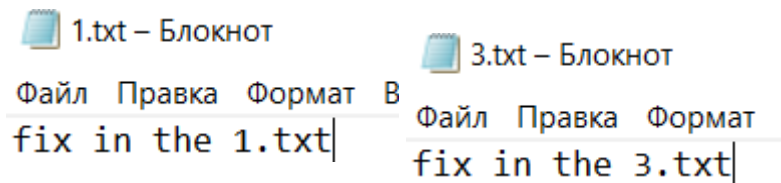


Рисунок 4.3 Изменения в файлах 1.txt и 3.txt

```
C:\Users\FonK\Desktop\ОПИ\labRab_3>git commit -m "fix 1.txt and 3.txt in branch_1"
[branch_1 ff76711] fix 1.txt and 3.txt in branch_1
3 files changed, 5 insertions(+), 2 deletions(-)
```

Рисунок 4.4 Коммит изменений

```
C:\Users\FonK\Desktop\ОПИ\labRab_3>git checkout branch_2
Switched to branch 'branch_2'
```

Рисунок 4.5 Переход на ветку branch\_2

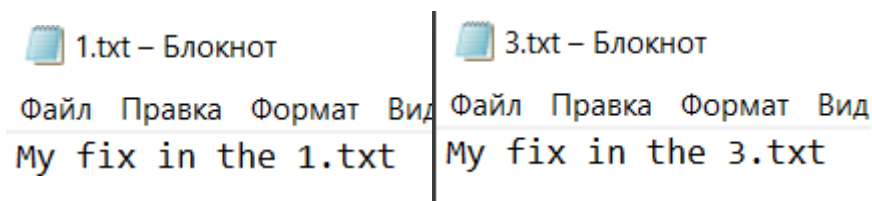


Рисунок 4.6 Изменения в файлах 1.txt и 3.txt

```
C:\Users\FonK\Desktop\ОПИ\labRab_3>git add .
C:\Users\FonK\Desktop\ОПИ\labRab_3>git commit -m "fix 1.txt and 3.txt in branch_2"
[branch_2 c947db5] fix 1.txt and 3.txt in branch_2
2 files changed, 2 insertions(+), 1 deletion(-)
```

Рисунок 4.7 Коммит изменений

5. 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\FonK\Desktop\ОПИ\labRab_3>git checkout branch_1
Switched to branch 'branch_1'

C:\Users\FonK\Desktop\ОПИ\labRab_3>git merge branch_2
Auto-merging 3.txt
CONFLICT (content): Merge conflict in 3.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Рисунок 5.1 Слияние веток

```
C:\Users\FonK\Desktop\ОПИ\labRab_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): vimdiff
```

Рисунок 5.2 Команда mergetool

```
C:\Users\FonK\Desktop\ОПИ\labRab_3>git status
On branch branch_1
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
  modified:   3.txt
```

Рисунок 5.3 Пуш ветки на уд. репозиторий

6. 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”:

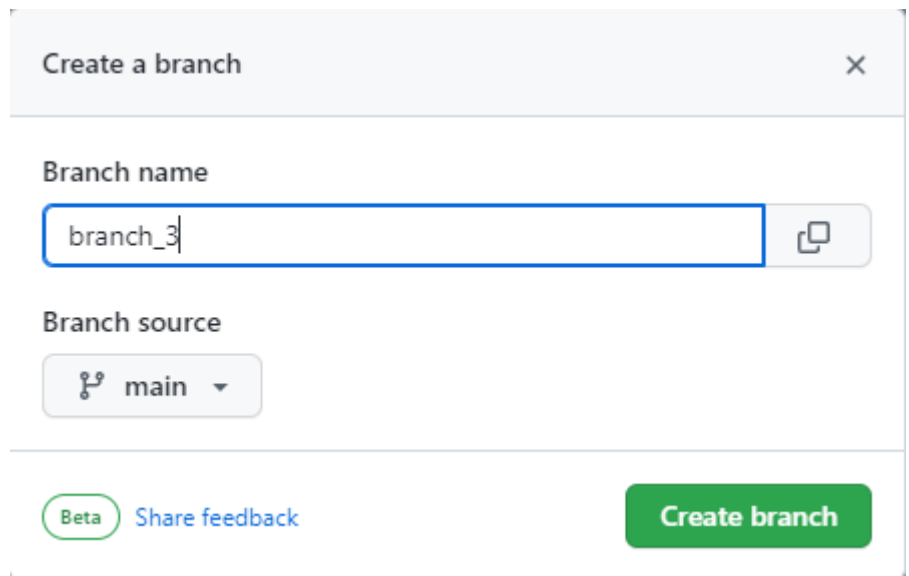


Рисунок 6.1 Создание уд. ветки branch\_3

```
C:\Users\FonK\Desktop\ОПИ\labRab_3>git fetch --all
From https://github.com/kvakaet/labRab_3
* [new branch]      branch_3    -> origin/branch_3
```

Рисунок 6.2 Загрузка содержимого из уд. репозитория перед  
отслеживанием

```
C:\Users\FonK\Desktop\ОПИ\labRab_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\FonK\Desktop\ОПИ\labRab_3>git branch
  branch_1
  branch_2
* branch_3
  main
```

Рисунок 6.3 Создание ветки отслеживания branch\_3

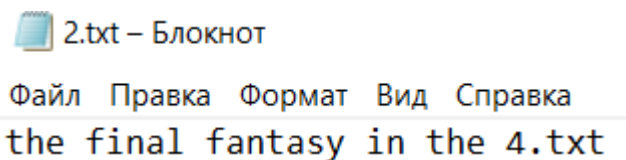


Рисунок 6.4 Добавление в 2.txt строку “the final fantasy in the 4.txt file” на  
ветке branch\_3



**Ссылка на репозиторий Git Hub:** [https://github.com/kvakaet/labRab\\_3.git](https://github.com/kvakaet/labRab_3.git)

## **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>`.



