

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

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №1.2

Исследование
основных возможностей
Git для работы с локальным репозиторием

Выполнил студент группы

ИВТ-б-о-21-1

Мальцев Н.А. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций, старший
преподаватель

Воронкин Р.А.

(подпись)

Ставрополь 2022

Основы ветвления в Git.

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

Порядок выполнения работы:

- 1) Изучен теоретический материал лабораторной работы.
- 2) Создан общедоступный репозиторий с лицензией MIT на Git Hub.

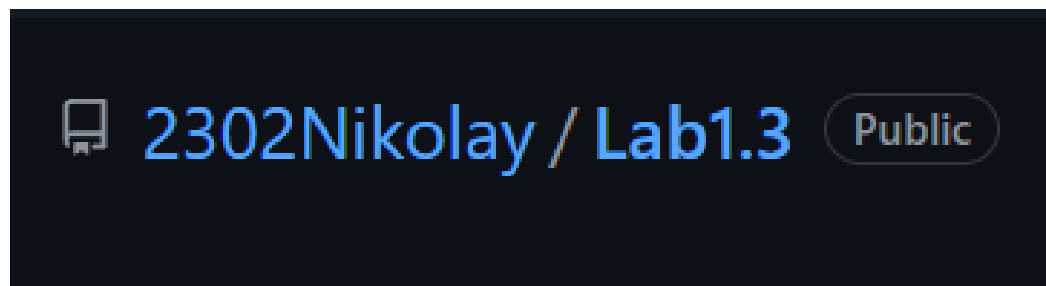


Рисунок 1. Созданный на Git Hub репозиторий

- 3) Создано три текстовых файла.

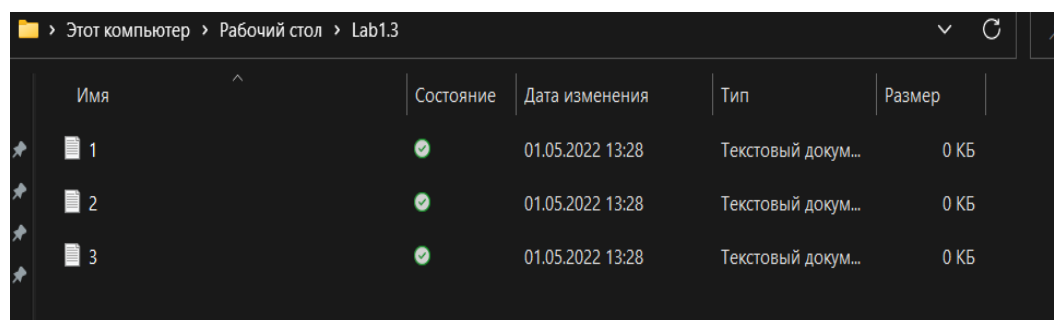


Рисунок 2. Созданные текстовые файлы

- 4) Проиндексировал первый файл (1.txt).

```
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git add 1.txt
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        2.txt
        3.txt
```

Рисунок 3. Проиндексировал файл

- 4.1) Сделал коммит с комментарием «add 1.txt file».

```
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git commit -m"add 1.txt file"
[master (root-commit) 5022d2d] add 1.txt file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> |
```

Рисунок 4. Коммит об индексации файла 1.txt

5) Аналогично проиндексировал файлы 2.txt и 3.txt.

```
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git add 2.txt
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git add 3.txt
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   2.txt
        new file:   3.txt
```

Рисунок 5. Проиндексировал оставшиеся файлы

6) Создал новую ветку my_first_branch.

7) Перешёл на новую ветку.

```
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git checkout my_first_branch
Switched to branch 'my_first_branch'
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> |
```

Рисунок 6. Переключение на ветку my_first_branch

8) Добавил новый файл и создал коммит в новой ветке.

```
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git add inbranch.txt
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git status
On branch my_first_branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   inbranch.txt

PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git commit -m"add inbranch.txt file on new branch"
[my_first_branch c2f27fa] add inbranch.txt file on new branch
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 inbranch.txt
```

Рисунок 7. Результат выполнения команд в командной строке

9) Создал новую ветку, перешёл в неё, изменил файл 1.txt, зафиксировал изменения и добавил коммит.

```
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git branch new_branch
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git checkout new_branch
Switched to branch 'new_branch'
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git status
On branch new_branch
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   1.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git add .
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git commit -m"Changed first file"
[new_branch cc5e77c] Changed first file
1 file changed, 1 insertion(+)
```

Рисунок 8. Результат выполнения команд в командной строке

10) Перешёл на ветку master и слил её с веткой my_first_branch.

```
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git checkout master
Switched to branch 'master'
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git merge my_first_branch
Updating 1b64a40..c2f27fa
Fast-forward
 inbranch.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 inbranch.txt
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> |
```

Рисунок 9. Результат выполнения команд в командной строке

11) Слил ветки master и new_branch.

```
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git merge new_branch
Merge made by the 'ort' strategy.
 1.txt | 1 +
 1 file changed, 1 insertion(+)
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> |
```

Рисунок 10. Результат слияния веток

12) Удалил ветки my_first_branch и new_branch.

```
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git branch -d my_first_branch
Deleted branch my_first_branch (was c2f27fa).
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git branch -d new_branch
Deleted branch new_branch (was cc5e77c).
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> |
```

Рисунок 11. Удаление веток

13) Создал ветки branch_1 и branch_2.

```
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git branch branch_1
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git branch branch_2
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> |
```

Рисунок 12. Создал две новые ветки

14) Перешёл на ветку branch_1 и изменил файлы 1.txt и 3.txt, зафиксировал изменения и создал коммит.

```
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git checkout branch_1
Switched to branch 'branch_1'
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git status
On branch branch_1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   1.txt
        modified:   3.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git add .
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git status
On branch branch_1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   1.txt
        modified:   3.txt

PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git commit -m"Changed first and third file"
[branch_1 74f0452] Changed first and third file
 2 files changed, 2 insertions(+), 1 deletion(-)
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> |
```

Рисунок 13. Изменил первый и третий файлы и создал коммит

15) Перешёл на ветку branch_2 и изменил файлы 1.txt и 3.txt, зафиксировал изменения и создал коммит.

```
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git checkout branch_2
Switched to branch 'branch_2'
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git status
On branch branch_2
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   1.txt
        modified:   3.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git add .
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git status
On branch branch_2
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   1.txt
        modified:   3.txt

PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git commit -m"Changed first and third files on branch_2"
[branch_2 5ea5cd9] Changed first and third files on branch_2
 2 files changed, 2 insertions(+), 1 deletion(-)
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> |
```

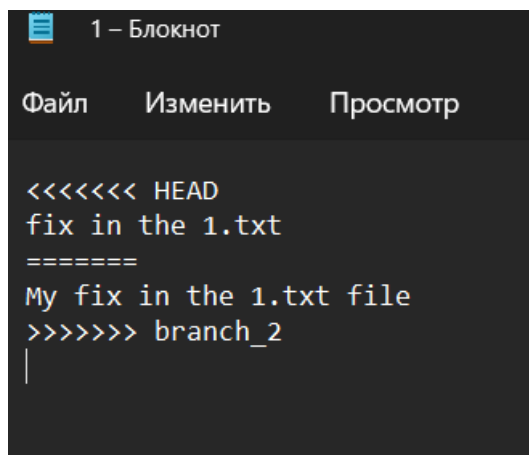
Рисунок 14. Изменил первый и третий файлы и создал коммит

16) Слил изменения в ветки branch_2 в ветку branch_1.

```
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git checkout branch_1
Switched to branch 'branch_1'
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\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.
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> |
```

Рисунок 15. Слияние двух новых веток

17) Конфликт файла 1.txt решил в ручном режиме.



```
1 – Блокнот
Файл  Изменить  Просмотр

<<<<<<< HEAD
fix in the 1.txt
=====
My fix in the 1.txt file
>>>>>>> branch_2
|
```

Рисунок 16. Конфликт в файле 1

18) Отправил ветку branch_1 на Git Hub.

```

PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git push origin branch_1
Enumerating objects: 21, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 12 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (18/18), 1.60 KiB | 820.00 KiB/s, done.
Total 18 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), done.
remote:
remote: Create a pull request for 'branch_1' on GitHub by visiting:
remote:   https://github.com/2302Nikolay/Lab1.3/pull/new/branch_1
remote:
To https://github.com/2302Nikolay/Lab1.3.git
 * [new branch]      branch_1 -> branch_1
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> |

```

Рисунок 17. Отправил ветку на удаленный репозиторий

19) На Git Hub создал ветку branch_3.

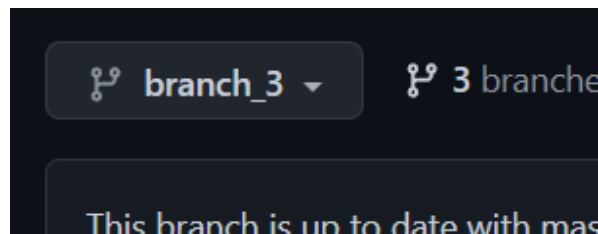


Рисунок 18. Создал ветку на git hub

20) В локальном репозитории создана ветка отслеживания удаленной ветки branch_3.

```

PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git checkout --track origin/branch_3
Switched to a new branch 'branch_3'
branch 'branch_3' set up to track 'origin/branch_3'.
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> |

```

Рисунок 19. Создание ветки отслеживания

21) Выполнил перемещение ветки master на ветку branch_2.

```

PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git checkout branch_2
Switched to branch 'branch_2'
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git rebase master
Current branch branch_2 is up to date.
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git checkout master
Switched to branch 'master'
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git merge branch_2
Updating b8b442e..5ea5cd9
Fast-forward
 1.txt | 2 +
 3.txt | 1 +
 2 files changed, 2 insertions(+), 1 deletion(-)
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> |

```

Рисунок 20. Результат выполнения команд

22) Отправил все изменения на удаленный репозиторий.

```
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> git push origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/2302Nikolay/Lab1.3.git
 1b64a40..5ea5cd9  master -> master
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> 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/2302Nikolay/Lab1.3/pull/new/branch_2
remote:
To https://github.com/2302Nikolay/Lab1.3.git
 * [new branch]      branch_2 -> branch_2
PS C:\Users\Николай Мальцев\OneDrive\Рабочий стол\Lab1.3> |
```

Рисунок 21. Результат выполнения команд

Ссылка на репозиторий Git Hub: <https://github.com/2302Nikolay/Lab1.3.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 локальной 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>`.

Вывод: в результате работы были исследованы базовые возможности по работе с локальными и удаленными ветками `Git`.