

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
**Федеральное государственное автономное образовательное учреждение
высшего образования**
«Северо-Кавказский федеральный университет»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №3

«Основы ветвления Git»

по дисциплине «Основы программной инженерии»

Выполнила:

Лазарева Дарья Олеговна,

2 курс, группа ПИЖ-б-о-20-1,

Проверил:

Доцент кафедры инфокоммуникаций,

Воронкин Роман Александрович

Ставрополь, 2021 г.

1. Добавление еще одного файла

```
C:\Users\79616\work3>git log --oneline
c8e6e15 (HEAD) Добавлен еще один файл
8475d90 Добавление файлов
916f4ac Initial commit
```

2. Перенос указателя «HEAD» на другую ветку

```
C:\Users\79616\work3>git checkout 8475d90
Warning: you are leaving 1 commit behind, not connected to
any of your branches:

c8e6e15 добавлен еще один файл

If you want to keep it by creating a new branch, this may be a good time
to do so with:

git branch <new-branch-name> c8e6e15

HEAD is now at 8475d90 Добавление файлов
```

```
C:\Users\79616\work3>git log --oneline
8475d90 (HEAD) Добавление файлов
916f4ac Initial commit
```

3. Возвращение ветки на место

```
C:\Users\79616\work3>git checkout c8e6e15
Previous HEAD position was 8475d90 Добавление файлов
HEAD is now at c8e6e15 Добавлен еще один файл

C:\Users\79616\work3>git log --oneline
c8e6e15 (HEAD) Добавлен еще один файл
8475d90 Добавление файлов
916f4ac Initial commit
```

4. Содержимое репозитория на момент отправки коммита

```
C:\Users\79616\work3>cd.git
C:\Users\79616\work3\.git>dir
Том в устройстве C имеет метку Windows
Серийный номер тома: CA66-A77C

Содержимое папки C:\Users\79616\work3\.git
20.09.2021  22:23                42 COMMIT_EDITMSG
20.09.2021  21:50            301 config
20.09.2021  21:50            73 description
20.09.2021  22:26            41 HEAD
20.09.2021  21:50      <DIR>      hooks
20.09.2021  22:26          577 index
20.09.2021  21:50      <DIR>      info
20.09.2021  21:50      <DIR>      logs
20.09.2021  22:23      <DIR>      objects
20.09.2021  21:50          112 packed-refs
20.09.2021  21:50      <DIR>      refs
                        6 файлов          1 146 байт
                        5 папок   27 428 044 800 байт свободно
```

5. Содержимое файла HEAD

```
C:\Users\79616\work3\.git>type HEAD
c8e6e1569850f8d96155778a5462b22efcfec2d4
```

6. Создание ветки my_first_branch

```
C:\Users\79616\work3>git branch my_first_branch
C:\Users\79616\work3>git log --oneline
c8e6e15 (HEAD, testing, my_first_branch) Добавлен еще один файл
8475d90 Добавление файлов
916f4ac Initial commit
```

7. Переключение на ветку my_first_branch

```
C:\Users\79616\work3>git checkout my_first_branch
Switched to branch 'my_first_branch'
```

```
C:\Users\79616\work3>git commit -a -m "made a change"
On branch my_first_branch
nothing to commit, working tree clean
```

8. Список всех имеющихся веток

```
C:\Users\79616\work3>git branch
main
* master
my_first_branch
```

9. Просмотр последнего коммита на каждой из веток

```
C:\Users\79616\work3>git branch -v
main          442630a Добавлены документы
* my_first_branch c8e6e15 Добавлен еще один файл
testing       c8e6e15 Добавлен еще один файл
```

10. Фильтрация веток по их слиянию/не слиянию

```
C:\Users\79616\work3>git branch --merged
* my_first_branch
testing

C:\Users\79616\work3>git branch --no-merged
main
```

```
C:\Users\79616\work3>git branch --no-merged
main

C:\Users\79616\work3>git branch -d main
warning: deleting branch 'main' that has been merged to
'refs/remotes/origin/main', but not yet merged to HEAD.
Deleted branch main (was 442630a).
```

11. Создание новой ветки и внесение изменений

```
C:\Users\79616\work3>git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   1.txt
        deleted:    4.txt

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\79616\work3>git add .

C:\Users\79616\work3>git commit -m "add 1.txt file"
[master b34d80d] add 1.txt file
 2 files changed, 1 insertion(+)
 delete mode 100644 4.txt

C:\Users\79616\work3>git status
On branch master
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:   2.txt
        modified:   3.txt

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\79616\work3>git add .

C:\Users\79616\work3>git commit -m "add 2.txt and 3.txt"
[master 307aea1] add 2.txt and 3.txt
 2 files changed, 2 insertions(+)
```

```
C:\Users\79616\work3>git checkout my_first_branch
Switched to branch 'my_first_branch'

C:\Users\79616\work3>git status
On branch my_first_branch
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        in_branch.txt.txt
```

12. Создание и переход на новую ветку new_branch

```
C:\Users\79616\work3>git checkout master
Switched to branch 'master'

C:\Users\79616\work3>git branch new_branch

C:\Users\79616\work3>git checkout new_branch
Switched to branch 'new_branch'
```

13. Слияние двух веток: master и new_branch

```
C:\Users\79616\work3>git checkout master
Switched to branch 'master'

C:\Users\79616\work3>git merge my_first_branch
Merge made by the 'recursive' strategy.
 in_branch.txt.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 in_branch.txt.txt

C:\Users\79616\work3>git checkout master
Already on 'master'

C:\Users\79616\work3>git merge new_branch
Merge made by the 'recursive' strategy.
 1.txt | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```

14. Удаление веток my_first_branch и new_branch

```
C:\Users\79616\work3>git branch -d my_first_branch
Deleted branch my_first_branch (was 5c6b295).

C:\Users\79616\work3>git branch -d new_branch
Deleted branch new_branch (was 23ad49d).
```

15. Создание веток branch_1 и branch_2

```
C:\Users\79616\work3>git branch branch_1
C:\Users\79616\work3>git branch branch_2
```

16. Внесение изменений в ветку branch_1

```
C:\Users\79616\work3>git checkout branch_1
Switched to branch 'branch_1'

C:\Users\79616\work3>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")

C:\Users\79616\work3>git add .

C:\Users\79616\work3>git commit -m "Добавлены изменения"
[branch_1 23de538] Добавлены изменения
 2 files changed, 2 insertions(+), 3 deletions(-)
```

17. Внесение изменений в ветку branch_2

```
C:\Users\79616\work3>git checkout branch_2
Switched to branch 'branch_2'

C:\Users\79616\work3>git add .

C:\Users\79616\work3>git commit -m "Добавление изменений"
[branch_2 b77fa6e] Добавление изменений
 2 files changed, 2 insertions(+), 3 deletions(-)
```

18. Слияние веток branch_1 и branch_2

```
C:\Users\79616\work3>git checkout branch_1
Switched to branch 'branch_1'

C:\Users\79616\work3>git merge branch_2
Auto-merging 3.txt
CONFLICT (content): Merge conflict in 3.txt
Auto-merging 1.txt
CONFLICT (content): Merge conflict in 1.txt
Automatic merge failed; fix conflicts and then commit the result.
```


19. Отправление ветки branch_1 на GitHub

```
C:\Users\79616\work3>git push origin branch_1
Enumerating objects: 30, done.
Counting objects: 100% (30/30), done.
Delta compression using up to 8 threads
Compressing objects: 100% (22/22), done.
Writing objects: 100% (30/30), 4.07 KiB | 1.02 MiB/s, done.
Total 30 (delta 9), reused 3 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (9/9), done.
remote:
remote: Create a pull request for 'branch_1' on GitHub by visiting:
remote:   https://github.com/LazarevaDarya/work3/pull/new/branch_1
remote:
To https://github.com/LazarevaDarya/work3.git
 * [new branch]      branch_1 -> branch_1
```

20. Создание ветки веток branch_3

```
C:\Users\79616\work3>git branch branch_3
C:\Users\79616\work3>git checkout branch_3
Switched to branch 'branch_3'
M       1.txt
M       3.txt
C:\Users\79616\work3>git push origin branch_3
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'branch_3' on GitHub by visiting:
remote:   https://github.com/LazarevaDarya/work3/pull/new/branch_3
remote:
To https://github.com/LazarevaDarya/work3.git
 * [new branch]      branch_3 -> branch_3
```

21. Отправление веток branch_2 и master на GitHub

```
C:\Users\79616\work3>git push origin branch_2
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 360 bytes | 360.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'branch_2' on GitHub by visiting:
remote:   https://github.com/LazarevaDarya/work3/pull/new/branch_2
remote:
To https://github.com/LazarevaDarya/work3.git
 * [new branch]      branch_2 -> branch_2

C:\Users\79616\work3>git push origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/LazarevaDarya/work3/pull/new/master
remote:
To https://github.com/LazarevaDarya/work3.git
 * [new branch]      master -> master
```


ОТВЕТЫ НА ВОПРОСЫ:

1. Что такое ветка? Ветка в Git — это простой перемещаемый указатель на коммиты. По умолчанию, имя основной ветки в Git — master . Как только вы начнёте создавать коммиты, ветка master будет всегда указывать на последний коммит.
2. Что такое HEAD? HEAD – это указатель, задача которого ссылаться на определенный коммит в репозитории. Во-первых, HEAD – это указатель на коммит в вашем репозитории, который станет родителем следующего коммита. Во-вторых, HEAD указывает на коммит, относительно которого будет создана рабочая копия во-время операции checkout. Другими словами, когда вы переключаетесь с ветки на ветку, используя операцию checkout, то в вашем репозитории указатель HEAD будет переключаться между последними коммитами выбираемых вами ветвей.
3. Способы создания веток. С помощью команды `git branch`
4. Как узнать текущую ветку? Он хранит специальный указатель HEAD. С помощью команды `git log --decorate` можно посмотреть, куда указывают указатели веток
5. Как переключаться между ветками? Для переключения на существующую ветку выполните команду `git checkout <название ветки>`
6. Что такое удаленная ветка? Удаленные ссылки – это ссылки (указатели) в удаленных репозиториях, включая ветки, теги и тд. Полный список удаленных ссылок можно получить с помощью команды `git ls-remote <name>` или `git remote show <remote>`
7. Что такое ветка отслеживания? Ветки слежения — это локальные ветки, которые напрямую связаны с удалённой веткой. Git перемещает их автоматически при любой коммуникации с удаленным репозиторием, чтобы гарантировать точное соответствие с ним.
8. Как создать ветку отслеживания? `git branch --track <name>`
9. Как отправить изменения из локальной ветки в удаленную ветку? Если у вас есть ветка, над которой вы хотите работать с кем-то ещё, вы можете

отправить её точно так же, как отправляли первую ветку. Команда `git push <remote> <branch>` :

10. В чем отличие команд `git fetch` и `git pull`? Для синхронизации ваших изменений с удаленным сервером выполните команду `git fetch <remote>`. Эта команда определяет какому серверу соответствует “origin” (в нашем случае это `git.ourcompany.com`), извлекает оттуда данные, которых у вас ещё нет, и обновляет локальную базу данных, сдвигая указатель `origin/master` на новую позицию. директории. Эта команда просто получает данные и позволяет вам самостоятельно сделать слияние. Тем не менее, существует команда `git pull`. `git pull` определит сервер и ветку, за которыми следит ваша текущая ветка, получит данные с этого сервера и затем попытается слить удалённую ветку.

11. Как удалить локальную и удаленную ветки? Вы можете удалить ветку на удалённом сервере используя параметр `--delete` для команды `git push` . Для удаления ветки `serverfix` на сервере, выполните следующую команду: `git push origin --delete serverfix`

12. Изучить модель ветвления `git-flow`. Какие основные типы веток присутствуют в модели `git-flow`? Как организована работа с ветками в модели `git-flow`? В чем недостатки `git-flow`? `Git-flow` — альтернативная модель ветвления `Git`, в которой используются функциональные ветки и несколько основных веток. В этом рабочем процессе для регистрации истории проекта вместо одной ветки `main` используются две ветки. В главной ветке `main` хранится официальная история релиза, а ветка разработки `develop` предназначена для объединения всех функций. Кроме того, для удобства рекомендуется присваивать всем коммитам в ветке `main` номер версии.

13. На прошлой лабораторной работе было задание выбрать одно из программных средств с GUI для работы с `Git`. Необходимо в рамках этого вопроса привести описание инструментов для работы с ветками `Git`, предоставляемых этим средством.