

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»**

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2

по дисциплине «Основы программной инженерии»

Выполнил студент группы ПИЖ-б-о-20-1

Ваньянц И.М. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2021

ХОД РАБОТЫ

```
C:\Users\Илья\NCFU\labs>git add 1.txt

C:\Users\Илья\NCFU\labs>git status
On branch main
Your branch is up to date with 'origin/main'.

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

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

C:\Users\Илья\NCFU\labs>git commit -m "1.txt committed"
[main 83ff656] 1.txt committed
1 file changed, 1 insertion(+)
create mode 100644 1.txt
```

Рисунок 1 – коммит первого файла

```
C:\Users\Илья\NCFU\labs>git add 2.txt

C:\Users\Илья\NCFU\labs>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   2.txt

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

C:\Users\Илья\NCFU\labs>git commit -m "2.txt committed"
[main 79289e6] 2.txt committed
1 file changed, 1 insertion(+)
create mode 100644 2.txt
```

Рисунок 2 – коммит второго файла

```
C:\Users\Илья\NCFU\labs>git commit -m "2.txt committed"
[main 79289e6] 2.txt committed
1 file changed, 1 insertion(+)
create mode 100644 2.txt

C:\Users\Илья\NCFU\labs>git add 3.txt

C:\Users\Илья\NCFU\labs>git commit -m "3.txt committed"
[main 918f391] 3.txt committed
1 file changed, 1 insertion(+)
create mode 100644 3.txt
```

Рисунок 3 – коммит третьего файла

```

C:\Users\Илья\NCFUlabs>git checkout 79289e6
Note: switching to '79289e6'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 79289e6 2.txt committed

```

Рисунок 4 – сдвиг HEAD на два коммита назад

```

C:\Users\Илья\NCFUlabs>cd .git
C:\Users\Илья\NCFUlabs\.git>dir
Том в устройстве C не имеет метки.
Серийный номер тома: 583C-FE44

Содержимое папки C:\Users\Илья\NCFUlabs\.git

24.11.2021  22:19                12 288 .COMMIT_EDITMSG.swo
14.11.2021  20:28                12 288 .COMMIT_EDITMSG.swp
24.11.2021  22:31                  7 COMMIT_EDITMSG
24.11.2021  22:41               365 config
14.11.2021  18:54                73 description
18.11.2021  20:28               100 FETCH_HEAD
24.11.2021  22:44                21 HEAD
14.11.2021  18:54             <DIR> hooks
24.11.2021  22:44             1 136 index
14.11.2021  18:54             <DIR> info
14.11.2021  18:54             <DIR> logs
24.11.2021  22:33             <DIR> objects
18.11.2021  20:28                41 ORIG_HEAD
14.11.2021  18:54             112 packed-refs
14.11.2021  18:54             <DIR> refs
                        10 файлов                26 431 байт
                        5 папок  83 885 707 264 байт свободно

C:\Users\Илья\NCFUlabs\.git>type head
ref: refs/heads/main

```

Рисунок 5 – просмотр содержимого репозитория

```

C:\Users\Илья\NCFUlabs>git branch testing

C:\Users\Илья\NCFUlabs>git log --oneline --decorate
2976411 (HEAD -> main, origin/main, origin/HEAD, testing) Revert "Revert "Revert "Revert "Revert "Revert "Revert "amend""""""
302d5c1 Revert "Revert "Revert "Revert "Revert "amend""""""
4431492 Revert "Revert "Revert "Revert "amend""""
44d5563 Revert "Revert "Revert "amend""
c575c73 Revert "Revert "amend""
a765bd2 Revert "amend"
2385553 amend
79289e6 (origin/master, temp, master) 2.txt committed
83ff656 1.txt committed
95870fd algoritm
b9dd193 Update README.md
bb5a78e commit for lab2
3d4d92e Index try
fe1b146 Try for amend
fa91007 Doc labs added
1794d2f Seventh commit
ab24b41 Sixth commit
f18223c fifth commit
46c606a fourth commit
f5d75c6 third commit
2dcf759 Update README.md
9aa8542 Initial commit

```

Рисунок 6 – создание ветви testing

```

C:\Users\Илья\NCFUlabs>git checkout testing
Switched to branch 'testing'

C:\Users\Илья\NCFUlabs>git commit -a -m "made a change"
On branch testing
nothing to commit, working tree clean

C:\Users\Илья\NCFUlabs>git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

```

Рисунок 7 – перемещение по ветвям

```

C:\Users\Илья\NCFUlabs>git checkout -b hotfix
Switched to a new branch 'hotfix'

C:\Users\Илья\NCFUlabs>vim index.html

C:\Users\Илья\NCFUlabs>git commit -a -m "fixed the broken email address"
On branch hotfix
nothing to commit, working tree clean

C:\Users\Илья\NCFUlabs>git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

C:\Users\Илья\NCFUlabs>git merge hotfix
Already up to date.

```

Рисунок 8 – переход на checkout и его слияние с master

```

C:\Users\Илья\NCFUlabs>git branch -d hotfix
Deleted branch hotfix (was 79289e6).

```

Рисунок 9 – удаление ветви hotfix

```

C:\Users\Илья\NCFUlabs>git branch
* main
  master
  temp
  testing

```

Рисунок 10 – список ветвей

```
C:\Users\Илья\NCFUlabs>git branch -v
* main      2976411 Revert "Revert "Revert "Revert "Revert "Revert "amend"
master      79289e6 2.txt committed
temp        79289e6 2.txt committed
testing     2976411 Revert "Revert "Revert "Revert "Revert "Revert "amend"
```

Рисунок 11 – просмотр коммитов в ветвях

```
C:\Users\Илья\NCFUlabs>git push origin main
Everything up-to-date
```

Рисунок 12 – отправка изменений

```
C:\Users\Илья\NCFUlabs>git branch -u origin/master
Branch 'main' set up to track remote branch 'master' from 'origin'.
```

Рисунок 13 – слежение за удаленной веткой

```
C:\Users\Илья\NCFUlabs>git branch -vv
* main      2976411 [origin/main] Revert "Revert "Revert "Revert "Revert "
master      79289e6 [origin/master] 2.txt committed
temp        79289e6 2.txt committed
testing     2976411 Revert "Revert "Revert "Revert "Revert "Revert "
```

Рисунок 14 – просмотр веток слежения

```
C:\Users\Илья\NCFUlabs>git push origin --delete main2
remote: This repository moved. Please use the new location:
remote:  https://github.com/vanjantsPIJ/NCFUlabs.git
To https://github.com/vanjantsPIJ/ncfulabs2.git
 - [deleted]          main2
```

Рисунок 15 – удаление ветви на удаленном репозитории

Switch branches/tags
×

Branches

Tags

Create branch: test_branch from 'main'

Рисунок 16 – создание новой ветки на github

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

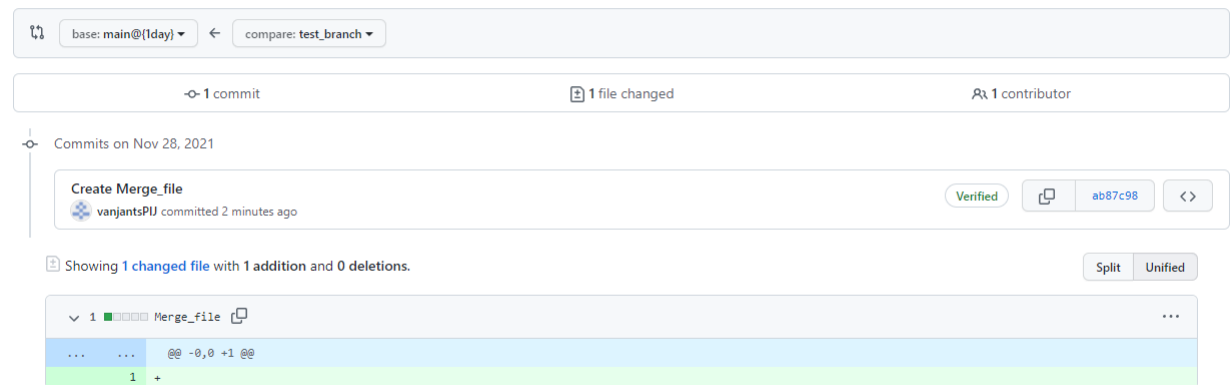


Рисунок 17 – создание pull request-a

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

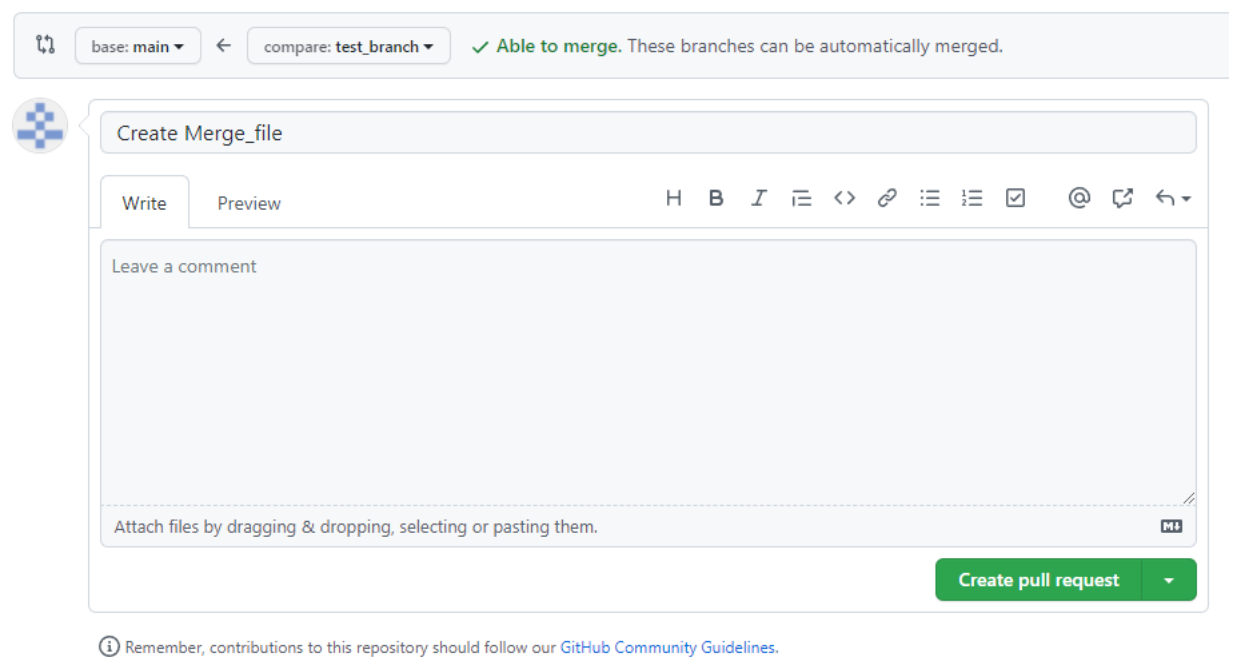


Рисунок 18 – отправка pull request-a

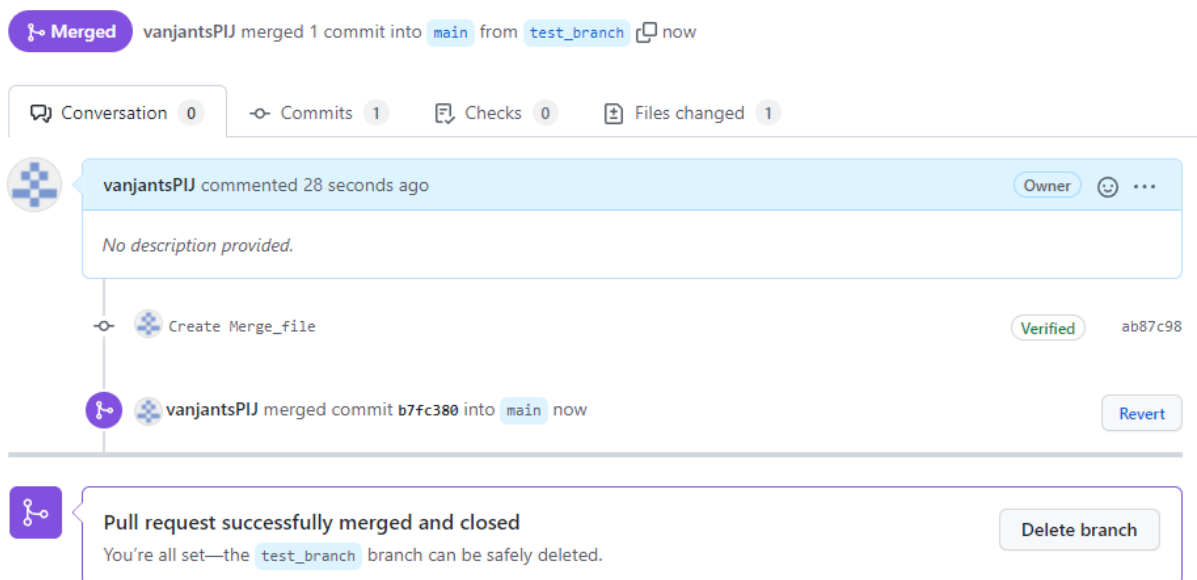


Рисунок 19 – слияние веток через pull request

```
C:\Users\Илья\NCFU\labs>git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

C:\Users\Илья\NCFU\labs>git rebase main
Successfully rebased and updated refs/heads/master.
```

Рисунок 20 – пример перебазирования

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Ветка – указатель, который перемещается по коммитам.
2. Head – указатель на коммит в вашем репозитории, который станет родителем следующего коммита.
3. Ветки в git могут создаваться: при помощи `git init` – создается ветка `master`; при помощи команды `git branch`; используя команду `git checkout -b` – сразу переключает на созданную ветку.
4. При помощи команды `git status` можно узнать, на какой ветке вы находитесь.
5. Переключение между ветками происходит при помощи команды `git checkout`
6. Ветка, которая находится на удалённом репозитории.
7. Ветка, которая отслеживает изменения другой ветки, и копирует эти изменения.
8. Отслеживать можно при помощи команды `git checkout -track`.
9. Отправить изменения на удалённую ветку можно при помощи команды – `git push origin имя_ветки`.
10. `Git fetch` получает с сервера все изменения, которых ещё нет, но не будет изменять состояние рабочей директории. `Git pull` – это команда `git fetch` только идущая в связке с командой `git merge`.
11. Чтобы удалить ветку на удалённом сервере используется команда `git push origin -delete имя_ветки`. Локальная ветка удаляется с командой `git push -d`.
12. В `git-flow` есть следующие типы веток: `feature branches`, `release branches`, `hotfix branches`. Модель работы с ветками следующая: 1. Из `master` создается ветка `develop`, из `develop` создаются ветки `feature`, когда разработка новой функциональности завершена, она объединяется с веткой `develop`, из `develop` создается ветка `release`, и когда ветка релиза готова, она объединяется с `develop` и `master`. Минусы `git-flow`: может замедлять работу, когда приходится отправлять большие пулл реквесты, когда итерацию нужно выполнить быстро, релизы сложно делать чаще раза в неделю, история проекта имеет кучу `merge` коммитов и затрудняет просмотр реальной работы.
13. `Gitk` показывает ветку коммитов на экране, что помогает более точно работать с коммитами.