

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет о лабораторной работе №2 по дисциплине основы программной
инженерии**

Выполнил:
Мамонтов Даниил
Вячеславович,
2 курс, группа ПИЖ-б-о-20-1,

Проверил:
Доцент кафедры
прикладной математики и
компьютерной безопасности,
Воронкин Р.А.

Отчет защищен с оценкой _____ Дата защиты _____

ВЫПОЛНЕНИЕ:

```
C:\Users\TEPLOX0\lab\lab2>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\TEPLOX0\lab\lab2>git commit -m "1.txt committed"
[main 52d2e50] 1.txt committed
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 lab2/1.txt
```

Рисунок 1.1 – Коммит первого файла

```
C:\Users\TEPLOX0\lab\lab2>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\TEPLOX0\lab\lab2>git commit -m "1.txt committed"
[main 52d2e50] 1.txt committed
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 lab2/1.txt
```

Рисунок 1.2 – Коммит второго файла

```
C:\Users\TEPLOX0\lab\lab2>git add 3.txt

C:\Users\TEPLOX0\lab\lab2>git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   3.txt

C:\Users\TEPLOX0\lab\lab2>git commit -m "3.txt committed"
[main c14f780] 3.txt committed
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 lab2/3.txt
```

Рисунок 1.3 – Коммит третьего файла

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 5848dc6 2.txt committed

Рисунок 1.4 – Сдвиг HEAD на два коммита назад

```
C:\Users\TEPLOX0\lab\.git>dir
Том в устройстве C не имеет метки.
Серийный номер тома: F295-F736

Содержимое папки C:\Users\TEPLOX0\lab\.git

18.01.2022  20:18                12 288 .COMMIT_EDITMSG.swp
18.01.2022  21:24                 15 COMMIT_EDITMSG
18.01.2022  17:32                293 config
18.01.2022  17:32                 73 description
18.01.2022  20:55                 94 FETCH_HEAD
18.01.2022  21:28                 41 HEAD
18.01.2022  17:32      <DIR>          hooks
18.01.2022  21:28      2 317 index
18.01.2022  17:32      <DIR>          info
18.01.2022  17:32      <DIR>          logs
18.01.2022  21:24      <DIR>          objects
18.01.2022  20:55                 41 ORIG_HEAD
18.01.2022  17:32                112 packed-refs
18.01.2022  17:32      <DIR>          refs
          9 файлов                15 274 байт
          5 папок 155 027 394 560 байт свободно
```

Рисунок 1.5 – Просмотр содержимого репозитория

```
C:\Users\TEPLOX0\lab>git branch testing
C:\Users\TEPLOX0\lab>git log --oneline --decorate
5848dc6 (HEAD, testing) 2.txt committed
52d2e50 1.txt committed
dbd08b5 (origin/main, origin/HEAD) Первая лабораторная завершена
0fe61e8 Вторая лабораторная завершена
2e7ec97 2
c3894fa 1
afd2b64 Зеркало
296146d Лабораторная 2 завершена
3af5800 commit
c152d14 Основные файлы
594c683 Лабораторная 2 завершена
99243c0 Merge branch 'main' of https://github.com/etozhekimm/lab1
```

Рисунок 1.6 – Создание ветви testing

```

C:\TEPLOX0\lab>git checkout testing
Switched to branch 'testing'

C:\TEPLOX0\lab>git commit -a -m "made a change"
On branch testing
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        git

nothing added to commit but untracked files present (use "git add" to track)

C:\TEPLOX0\lab>git add .

C:\TEPLOX0\lab>git commit -a -m "made a change"
[testing fd9c426] made a change
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 git

C:\TEPLOX0\lab>git checkout master
Switched to a new branch 'master'
Branch 'master' set up to track remote branch 'master' from 'origin'.

```

Рисунок 1.7 – Перемещение по ветвям

```

C:\TEPLOX0\lab>git checkout -b hotfix
Switched to a new branch 'hotfix'

C:\TEPLOX0\lab>vim index.html

C:\TEPLOX0\lab>git commit -a -m "fixed the broken email address"
On branch hotfix
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

nothing added to commit but untracked files present (use "git add" to track)

C:\TEPLOX0\lab>git add .

C:\TEPLOX0\lab>git commit -a -m "fixed the broken email address"
[hotfix 3011bcf] fixed the broken email address
 1 file changed, 2 insertions(+)
 create mode 100644 index.html

C:\TEPLOX0\lab>git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

C:\TEPLOX0\lab>git merge hotfix
Updating dbd08b5..3011bcf
Fast-forward
 index.html | 2 ++
 1 file changed, 2 insertions(+)

```

Рисунок 1.8 – Переход на checkout и его слияние с master

```
C:\TEPLOXO\lab>git branch -d hotfix
Deleted branch hotfix (was 3011bcf).

C:\TEPLOXO\lab>git branch
  main
* master
  testing
```

Рисунок 1.9 – Удаление ветви hotfix и список ветвей

```
C:\TEPLOXO\lab>git branch -v
  main      dbd08b5 Первая лабораторная завершена
* master    3011bcf [ahead 1] fixed the broken email address
  testing   fd9c426 made a change
```

Рисунок 1.10 – Просмотр последних коммитов в ветвях

```
C:\Users\TEPLOXO\lab\lab2>git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (10/10), 766 bytes | 191.00 KiB/s, done.
Total 10 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
remote: This repository moved. Please use the new location:
remote:  https://github.com/etozhekimm/lab.git
To https://github.com/etozhekimm/lab1
   dbd08b5..c14f780  main -> main
```

Рисунок 1.11 – Отправка изменений

```
C:\Users\TEPLOXO\lab\lab2>git branch -u origin/master
Branch 'hotfix' set up to track remote branch 'master' from 'origin'.
```

Рисунок 1.12 – Слежение за удаленной веткой

```
C:\Users\TEPLOXO\lab\lab2>git branch -vv
* hotfix    dbd08b5 [origin/master] Первая лабораторная завершена
  main      c14f780 [origin/main] 3.txt committed
  master    dbd08b5 [origin/master] Первая лабораторная завершена
  testing   5848dc6 2.txt committed
```

Рисунок 1.13 – Просмотр веток слежения

```
C:\Users\TEPLOXO\lab\lab2>git push origin --delete main2
error: unable to delete 'main2': remote ref does not exist
error: failed to push some refs to 'https://github.com/etozhekimm/lab1'
```

Рисунок 1.14 – Удаление ветви на удаленном репозитории

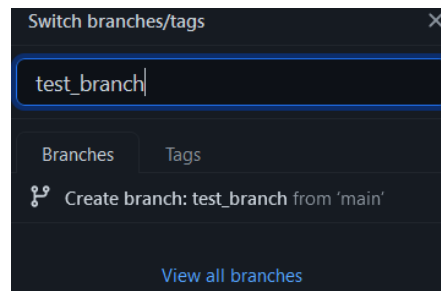


Рисунок 1.15 – Создание новой ветки на github

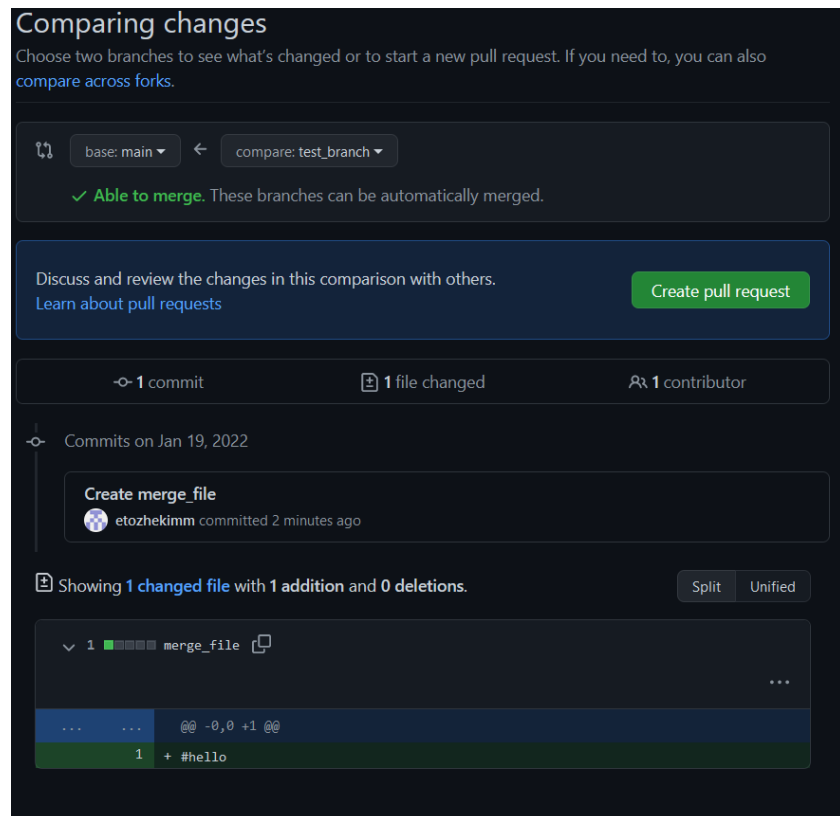


Рисунок 1.16 – Создание pull request-a

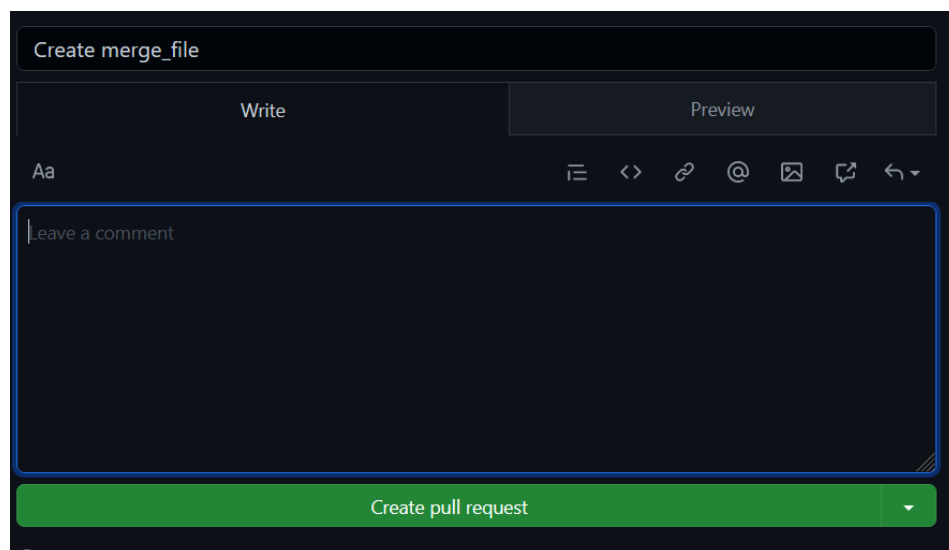


Рисунок 1.17 – Отправка pull request-a

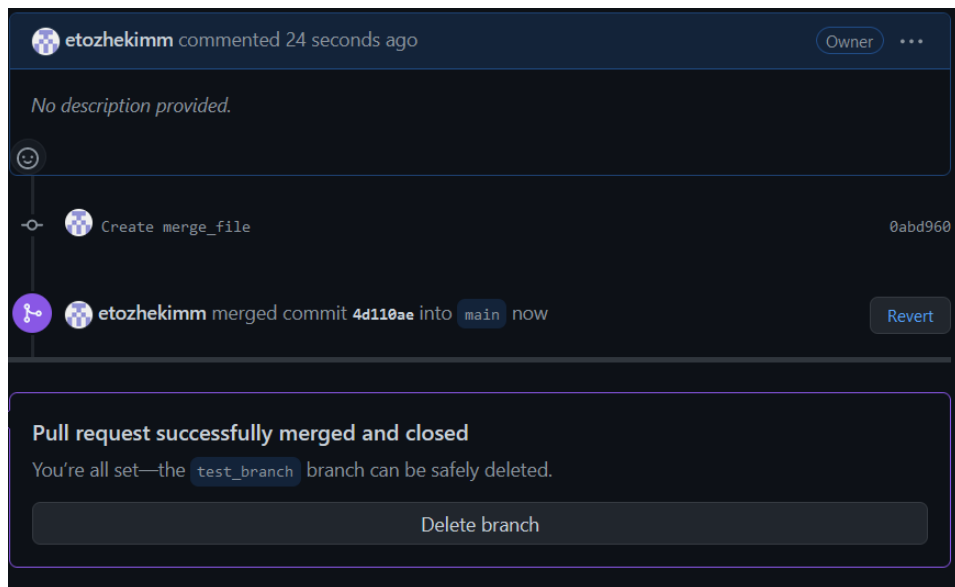


Рисунок 1.18 – Слияние веток через pull request

```
C:\Users\TEPLOX0\lab\lab2>git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

C:\Users\TEPLOX0\lab\lab2>git rebase main
Successfully rebased and updated refs/heads/master.
```

Рисунок 1.19 – Пример перебазирования

Ответы на 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.