

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

**«Исследование возможностей Git для работы с локальными
репозиториями»**

**ОТЧЕТ
по лабораторной работе №2
дисциплины
«Основы программной инженерии»**

Выполнила:

Кувшин Ирина Анатольевна

2 курс, группа ПИЖ-б-о-21-1,

09.03.04 «Программная инженерия»,

направленность (профиль) «Разработка

и сопровождение программного

обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

Лабораторная работа 1.2

Исследование возможностей Git для работы с локальными репозиториями

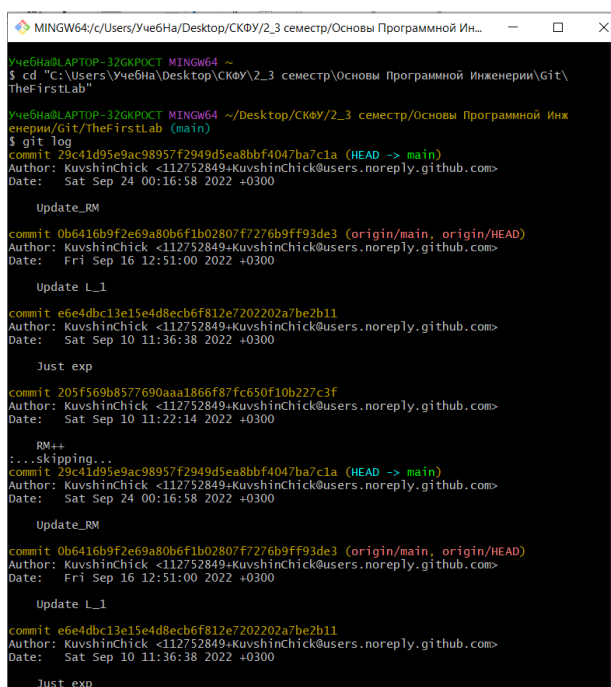
Цель работы: исследовать базовые возможности системы контроля версий Git для работы с локальными репозиториями.

Ссылка на репозиторий: <https://github.com/KuvshinChick/TheFirstLab.git>

Ход работы:

1. Просмотр истории коммитов

(8. Просмотреть историю (журнал) хранилища командой `git log`. Например, с помощью команды `git log --graph --pretty=oneline --abbrev-commit`. Добавить скриншот консоли с выводом в отчет по лабораторной работе.)



```
MINGW64~/Users/УчебНа/Desktop/СкФУ/2_3 семестр/Основы Программной Инж...
УчебНа@LAPTOP-32GKPOCT MINGW64 ~
$ cd "C:\Users\УчебНа\Desktop\СкФУ\2_3 семестр\Основы Программной Инж...
TheFirstLab"
УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СкФУ/2_3 семестр/Основы Программной Инж...
TheFirstLab (main)
$ git log
commit 29c41d95e9ac98957f2949d5ea8bbf4047ba7c1a (HEAD -> main)
Author: KuvshinChick <112752849+KuvshinChick@users.noreply.github.com>
Date: Sat Sep 24 00:16:58 2022 +0300

    Update_RM

commit 0b6416b9f2e69a80b6f1b02807f7276b9ff93de3 (origin/main, origin/HEAD)
Author: KuvshinChick <112752849+KuvshinChick@users.noreply.github.com>
Date: Fri Sep 16 12:51:00 2022 +0300

    Update_L_1

commit e6e4dbc13e15e4d8ecb6f812e7202202a7be2b11
Author: KuvshinChick <112752849+KuvshinChick@users.noreply.github.com>
Date: Sat Sep 10 11:36:38 2022 +0300

    Just exp

commit 205f569b8577690aaa1866f87fc650f10b227c3f
Author: KuvshinChick <112752849+KuvshinChick@users.noreply.github.com>
Date: Sat Sep 10 11:22:14 2022 +0300

    RM++
...skipping...
commit 29c41d95e9ac98957f2949d5ea8bbf4047ba7c1a (HEAD -> main)
Author: KuvshinChick <112752849+KuvshinChick@users.noreply.github.com>
Date: Sat Sep 24 00:16:58 2022 +0300

    Update_RM

commit 0b6416b9f2e69a80b6f1b02807f7276b9ff93de3 (origin/main, origin/HEAD)
Author: KuvshinChick <112752849+KuvshinChick@users.noreply.github.com>
Date: Fri Sep 16 12:51:00 2022 +0300

    Update_L_1

commit e6e4dbc13e15e4d8ecb6f812e7202202a7be2b11
Author: KuvshinChick <112752849+KuvshinChick@users.noreply.github.com>
Date: Sat Sep 10 11:36:38 2022 +0300

    Just exp
```

Рисунок 2.1 – использование команды `git log`



```
УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СкФУ/2_3 семестр/Основы Программной Инженерии
/Git/TheFirstLab (main)
$ git log -3
commit 29c41d95e9ac98957f2949d5ea8bbf4047ba7c1a (HEAD -> main)
Author: KuvshinChick <112752849+KuvshinChick@users.noreply.github.com>
Date: Sat Sep 24 00:16:58 2022 +0300

    Update_RM

commit 0b6416b9f2e69a80b6f1b02807f7276b9ff93de3 (origin/main, origin/HEAD)
Author: KuvshinChick <112752849+KuvshinChick@users.noreply.github.com>
Date: Fri Sep 16 12:51:00 2022 +0300

    Update_L_1

commit e6e4dbc13e15e4d8ecb6f812e7202202a7be2b11
Author: KuvshinChick <112752849+KuvshinChick@users.noreply.github.com>
Date: Sat Sep 10 11:36:38 2022 +0300

    Just exp
УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СкФУ/2_3 семестр/Основы Программной Инженерии
/Git/TheFirstLab (main)
$
```

Рисунок 2.2 – вывод последних трех коммитов

```

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии
/Git/TheFirstLab (main)
$ git log -p -1
commit 29c41d95e9ac98957f2949d5ea8bbf4047ba7c1a (HEAD -> main)
Author: KuvshinChick <112752849+KuvshinChick@users.noreply.github.com>
Date: Sat Sep 24 00:16:58 2022 +0300

    Update_RM

diff --git a/README.md b/README.md
index beb42ae..54f6f3e 100644
--- a/README.md
+++ b/README.md
@@ -2,4 +2,4 @@
    Кувшин Ирина Анатольевна
    ПИЖ-6-о-21-1 - 1 ПГ

-Лабораторная работа №1
+Лабораторная работа №1 + №2

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии
/Git/TheFirstLab (main)
$

```

Рисунок 2.3 – вывод патчей(разницы) последнего коммита

```

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии
/Git/TheFirstLab (main)
$ git log --pretty=oneline
29c41d95e9ac98957f2949d5ea8bbf4047ba7c1a (HEAD -> main) Update_RM
0b6416b9f2e69a80b6f1b02807f7276b9ff93de3 (origin/main, origin/HEAD) Update L_1
e6e4dbc13e15e4d8ecb6f812e7202202a7be2b11 Just exp
205f569b8577690aaa1866f87fc650f10b227c3f RM++
2794fd2c18d941cc417b8d73ec9d30460aa46942 Folder 'doc'
cc98fcb3d5e97e09a720bb439a1347146d2b2c4c + Otchet PDF
fd6fb385e2a69399828d0938714744cd4e491bf5 RM+
f7988ebb006bd80cf5131580d70e9b0c1d49d959 deleted otchet
ef9eb6a836e823db537fc4e1e1de027d5d2b60b6 +Otchet
7d38aae26bfd997aa387a4ddb3aef86ff74a1baa RM changed 2.0
669f53fe0c1994c429dff9e5e2739bd9c2eee67 RM was changed
455c13378461ad7a097bab44d76cce2e6efb2923 Add some nums
1190e3334b68f67174ee949e5f7a90bf2ee6ddee Add text 'Пижожж'
d60b1e2e4633ef107949c10ca216e4b37b7d91f4 2 commit C#
00623ed0cc9a55856c992309e9eb8cb7eb34ec66 b
14738512cc003cc5e7b6bce17d250125a41a8453 2 коммит RM
ab06b0d7bac85d1e4cdb36ce41f26b0dc1afdc created commit in RM
2907d8184369d614a93f43ab4239862a90692838 Initial commit

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии
/Git/TheFirstLab (main)
$

```

Рисунок 2.4 – использование команды git log --pretty=oneline

2. Просмотр содержимого коммитов

(9. Просмотреть содержимое коммитов командой git show <ref> , где <ref> :

HEAD : последний коммит;

HEAD~1 : предпоследний коммит (и т. д.);

b34a0e : коммит с указанным хэшем.

Отобразите результаты работы этих команд в отчете.)

```

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии
/Git/TheFirstLab (main)
$ git show HEAD
commit 29c41d95e9ac98957f2949d5ea8bbf4047ba7c1a (HEAD -> main)
Author: KuvshinChick <112752849+KuvshinChick@users.noreply.github.com>
Date: Sat Sep 24 00:16:58 2022 +0300

    Update_RM

diff --git a/README.md b/README.md
index beb42ae..54f6f3e 100644
--- a/README.md
+++ b/README.md
@@ -2,4 +2,4 @@
    Кувшин Ирина Анатольевна
    ПИЖ-6-о-21-1 - 1 ПГ

-Лабораторная работа №1
+Лабораторная работа №1 + №2

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии
/Git/TheFirstLab (main)
$

```

Рисунок 2.5 – просмотр содержимого последнего коммита

```

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии
/Git/TheFirstLab (main)
$ git show HEAD~2
commit e6e4dbc13e15e4d8ecb6f812e7202202a7be2b11
Author: KuvshinChick <112752849+KuvshinChick@users.noreply.github.com>
Date: Sat Sep 10 11:36:38 2022 +0300

    Just exp

diff --git a/README.md b/README.md
index 16ed593..beb42ae 100644
--- a/README.md
+++ b/README.md
@@ -1,5 +1,5 @@
    # TheFirstLab
    Кувшин Ирина Анатольевна
-ПИЖ-6-о-21-1 (1 ПГ)
+ПИЖ-6-о-21-1 - 1 ПГ

-Лабораторная работа №1
\ No newline at end of file
+Лабораторная работа №1

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии
/Git/TheFirstLab (main)
$

```

Рисунок 2.6 – просмотр содержимого пред – пред последнего коммита

```

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии
/Git/TheFirstLab (main)
$ git show ab06b0d7bac85d1e4cdb36ce41f26b0dc1afdc
commit ab06b0d7bac85d1e4cdb36ce41f26b0dc1afdc
Author: KuvshinChick <kuvshinirina75@gmail.com>
Date: Fri Sep 9 23:30:14 2022 +0300

    created commit in RM

diff --git a/README.md b/README.md
index c4ef3e6..6c49945 100644
--- a/README.md
+++ b/README.md
@@ -1 +1,4 @@
-# TheFirstLab
\ No newline at end of file
+# TheFirstLab
+Кувшин Ирина Анатольевна
+ПИЖ-6-о-21-1 (1)
+dgr

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии
/Git/TheFirstLab (main)
$

```

Рисунок 2.7 – просмотр содержимого по Хэшу

3. Откат к заданной версии:

(10. Освойте возможность отката к заданной версии.

10.1. Удалите весь код из одного из файлов программы репозитория, например main.cpp, и сохраните этот файл.

10.2. Удалите все несохраненные изменения в файле командой: `git checkout -- <имя_файла>`, например `git checkout -- main.cpp`.

10.3. Повторите пункт 10.1 и сделайте коммит.

10.4. Откатить состояние хранилища к предыдущей версии командой: `git reset --hard HEAD~1`.

Сделайте выводы об изменении содержимого выбранного Вами файла программы после выполнения пунктов 10.1–10.4. Отрадите эти выводы в отчете.)

```
Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии
/Git/TheFirstLab (main)
$ git checkout -- README.md

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии
/Git/TheFirstLab (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

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:   README.md

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

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии
/Git/TheFirstLab (main)
$ git add *

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии
/Git/TheFirstLab (main)
$ git commit -m "-Inf_in_README.md"
[main e8fa948] -Inf_in_README.md
1 file changed, 4 deletions(-)

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии
/Git/TheFirstLab (main)
$ git reset --hard HEAD~1
HEAD is now at 29c41d9 Update_RM

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab
(main)
$
```

Рисунок 2.8 – Выполнение команд

Выводы:

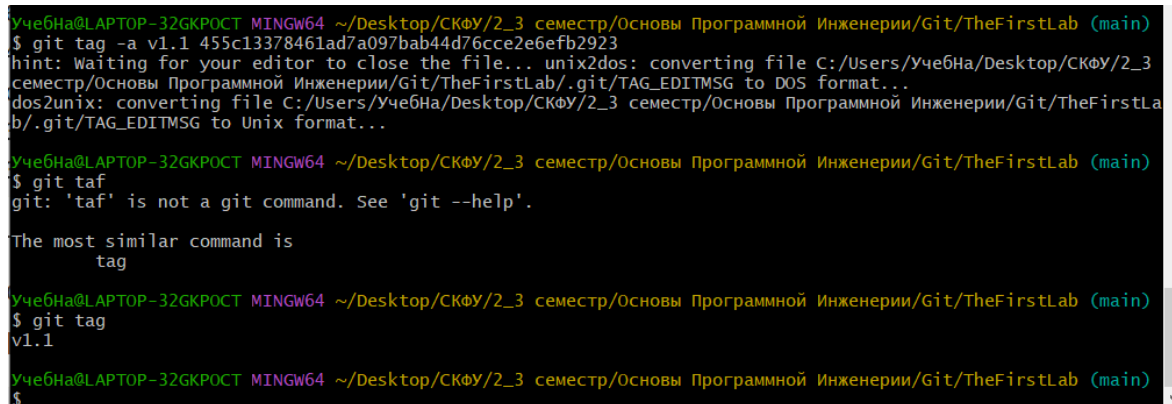
- в пункте 10.1 было произведено удаление данных из файла README.md
- в пункте 10.2 было произведено восстановление данных файла README.md за счет того, что несохраненное изменение (с удалением данных из п.10.1) было удалено командой `git checkout --`
- в пункте 10.3 было произведено удаление данных из файла README.md и произведен коммит
- в пункте 10.4 был произведен возврат к состоянию хранилища предыдущей версии командой `git reset --hard HEAD~1` => было произведено восстановление данных файла README.md

4. Работа с тэгами

(7. Напишите небольшую программу на выбранном Вами языке программирования. Фиксируйте изменения при написании программы в локальном репозитории. Должно быть сделано не менее 7 коммитов, отмеченных не менее 3 тэгами.)

Git использует два основных типа тегов: легковесные и аннотированные. Легковесный тег — это что-то очень похожее на ветку, которая не изменяется — просто указатель на определённый коммит. А вот аннотированные теги хранятся в базе данных Git как полноценные объекты. Они имеют контрольную сумму, содержат имя автора, его e-mail и дату создания, имеют комментарий и могут быть подписаны и проверены с помощью GNU Privacy Guard (GPG).

```
$ git tag -a v1.1 455c13378461ad7a097bab44d76cce2e6efb2923
```



```
Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (main)
$ git tag -a v1.1 455c13378461ad7a097bab44d76cce2e6efb2923
hint: Waiting for your editor to close the file... unix2dos: converting file C:/Users/Учебна/Desktop/СКФУ/2_3
семестр/Основы Программной Инженерии/Git/TheFirstLab/.git/TAG_EDITMSG to DOS format...
dos2unix: converting file C:/Users/Учебна/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLa
b/.git/TAG_EDITMSG to Unix format...

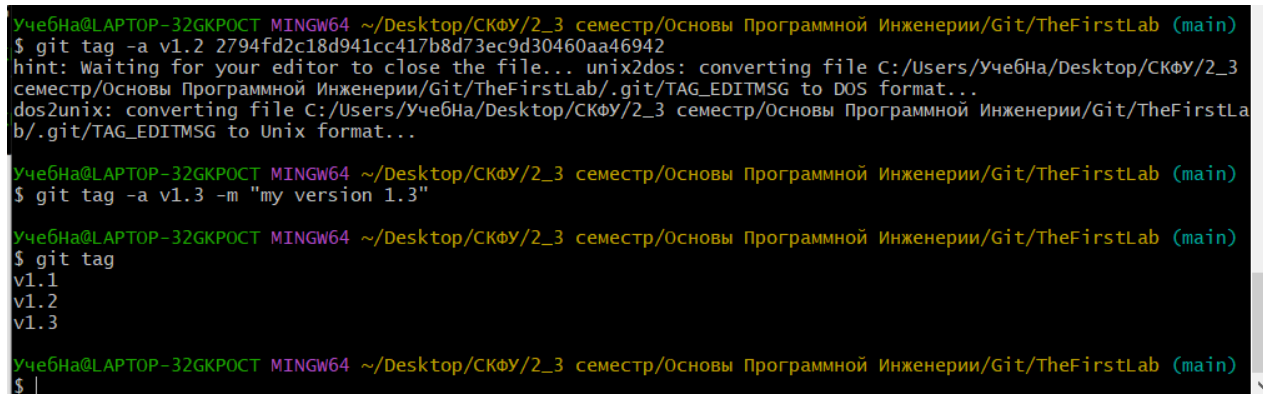
Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (main)
$ git taf
git: 'taf' is not a git command. See 'git --help'.

The most similar command is
    tag

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (main)
$ git tag
v1.1

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (main)
$
```

Рисунок 2.9 – Создание аннотированного тега по хэшу



```
Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (main)
$ git tag -a v1.2 2794fd2c18d941cc417b8d73ec9d30460aa46942
hint: Waiting for your editor to close the file... unix2dos: converting file C:/Users/Учебна/Desktop/СКФУ/2_3
семестр/Основы Программной Инженерии/Git/TheFirstLab/.git/TAG_EDITMSG to DOS format...
dos2unix: converting file C:/Users/Учебна/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLa
b/.git/TAG_EDITMSG to Unix format...

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (main)
$ git tag -a v1.3 -m "my version 1.3"

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (main)
$ git tag
v1.1
v1.2
v1.3

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (main)
$
```

Рисунок 2.10 – Создание двух тегов

```

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (main)
$ git show v1.1
tag v1.1
Tagger: KuvshinChick <112752849+KuvshinChick@users.noreply.github.com>
Date: Sat Sep 24 09:37:31 2022 +0300

The last change in C# file

commit 455c13378461ad7a097bab44d76cce2e6efb2923 (tag: v1.1)
Author: KuvshinChick <kuvshinirina75@gmail.com>
Date: Sat Sep 10 00:30:57 2022 +0300

    Add some nums

diff --git a/ConsoleApp1/ConsoleApp1/Program.cs b/ConsoleApp1/ConsoleApp1/Program.cs
index af56b89..05b2944 100644
--- a/ConsoleApp1/ConsoleApp1/Program.cs
+++ b/ConsoleApp1/ConsoleApp1/Program.cs
@@ -8,6 +8,7 @@ namespace ConsoleApp1
 {
     Console.WriteLine("Hello World!");
+    Console.WriteLine("ПИЖЖЖЖЖЖЖЖЖЖ");
+    Console.WriteLine('007');
 }
 }

```

Рисунок 2.11 – Просмотр данных тега v1.1 с коммитом

5. Создание зеркала (GitLab)

```

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 337 bytes | 337.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/KuvshinChick/TheFirstLab.git
   0b6416b..29c41d9  main -> main

```

Рисунок 2.12 – Отправка изменений в удаленный реп

```

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (main)
$ git fetch --prune

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (main)
$ git push --prune https://gitlab.com/KuvshinChick/theseclab.git
Enumerating objects: 64, done.
Counting objects: 100% (64/64), done.
Delta compression using up to 8 threads
Compressing objects: 100% (61/61), done.
Writing objects: 100% (64/64), 2.61 MiB | 959.00 KiB/s, done.
Total 64 (delta 25), reused 4 (delta 0), pack-reused 0
To https://gitlab.com/KuvshinChick/theseclab.git
 * [new branch]      main -> main

```

Рисунок 2.13 – отправка содержимого локального репозитория в реп GitLab



We stand with Ukraine

We have a commitment to our customers to provide a trusted service. In response, we are suspending new user sign-ups from Russia and Belarus.

[Learn more](#)

Вопросы:

1. Как выполнить историю коммитов в Git? Какие существуют дополнительные опции для просмотра истории коммитов?

- За просмотр истории коммитов отвечает команда `git log`.

Показывает все коммиты от новых к старым. Для каждого коммита выводится

- хэш
- автор
- дата
- сообщение (commit message)

- `git log -p`, расширенный вывод истории

Выводит то же, что и `git log`, но еще и с изменениями в файлах

○ `--stat` показывает сокращённую статистику для коммитов, например изменённые файлы и количество добавленных/удалённых строк в каждом из них;

- `-n` показывает `n` последних коммитов;

○ Следующей действительно полезной опцией является `--pretty`. Эта опция меняет формат вывода. Существует несколько встроенных вариантов отображения. Опция `oneline` выводит каждый коммит в одну строку, что может быть очень удобным если вы просматриваете большое количество коммитов. К тому же, опции `short`, `full` и `fuller` делают вывод приблизительно в том же формате, но с меньшим или большим количеством информации соответственно:

```
$ git log --pretty=oneline
```

```
ca82a6dff817ec66f44342007202690a93763949 Change version number
085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7 Remove unnecessary test
a11bef06a3f659402fe7563abf99ad00de2209e6 Initial commit
```

○ Наиболее интересной опцией является `format`, которая позволяет указать формат для вывода информации. Особенно это может быть полезным когда вы хотите сгенерировать вывод для автоматического анализа — так как вы указываете формат явно, он не будет изменен даже после обновления Git:

```
$ git log --pretty=format:"%h - %an, %ar : %s"
```

```
ca82a6d - Scott Chacon, 6 years ago : Change version number
085bb3b - Scott Chacon, 6 years ago : Remove unnecessary test
a11bef0 - Scott Chacon, 6 years ago : Initial commit
```

Опция	Описание
<code>-p</code>	Показывает патч для каждого коммита.
<code>--stat</code>	Показывает статистику измененных файлов для каждого коммита.
<code>--shortstat</code>	Отображает только строку с количеством изменений/вставок/удалений для команды <code>--stat</code> .
<code>--name-only</code>	Показывает список измененных файлов после информации о коммите.
<code>--name-status</code>	Показывает список файлов, которые добавлены/изменены/удалены.
<code>--abbrev-commit</code>	Показывает только несколько символов SHA-1 чек-суммы вместо всех 40.
<code>--relative-date</code>	Отображает дату в относительном формате (например, «2 weeks ago») вместо стандартного формата даты.
<code>--graph</code>	Отображает ASCII граф с ветвлениями и историей слияний.
<code>--pretty</code>	Показывает коммиты в альтернативном формате. Возможные варианты опций: <code>oneline</code> , <code>short</code> , <code>full</code> , <code>fuller</code> и <code>format</code> (с помощью последней можно указать свой формат).
<code>--oneline</code>	Сокращение для одновременного использования опций <code>--pretty=oneline</code> <code>--abbrev-commit</code> .

2. Как ограничить вывод при просмотре истории коммитов?

- В действительности вы можете использовать `-<n>`, где `n` — это любое натуральное число и представляет собой `n` последних коммитов.
- Однако, опции для ограничения вывода по времени, такие как `--since` и `--until`, являются очень удобными. Например, следующая команда покажет список коммитов, сделанных за последние две недели:
`$ git log --since=2.weeks`
- Следующим действительно полезным фильтром является опция `-S`, которая принимает аргумент в виде строки и показывает только те коммиты, в которых изменение в коде повлекло за собой добавление или удаление этой строки. Например, если вы хотите найти последний коммит, который добавил или удалил вызов определенной функции, вы можете запустить команду:
`$ git log -S function_name`
- Последней полезной опцией, которую принимает команда `git log` как фильтр, является путь. Если вы укажете каталог или имя файла, вы ограничите вывод только теми коммитами, в которых были изменения этих файлов. Эта опция всегда указывается последней после двойного тире (`--`), чтобы отделить пути от опций:
`$ git log -- path/to/file`

В нижеприведенной таблице Вы можете увидеть эти и другие распространенные опции.

Опция	Описание
<code>-(n)</code>	Показывает только последние <code>n</code> коммитов.
<code>--since, --after</code>	Показывает только те коммиты, которые были сделаны после указанной даты.
<code>--until, --before</code>	Показывает только те коммиты, которые были сделаны до указанной даты.
<code>--author</code>	Показывает только те коммиты, в которых запись <code>author</code> совпадает с указанной строкой.
<code>--committer</code>	Показывает только те коммиты, в которых запись <code>committer</code> совпадает с указанной строкой.
<code>--grep</code>	Показывает только коммиты, сообщение которых содержит указанную строку.
<code>-S</code>	Показывает только коммиты, в которых изменение в коде повлекло за собой добавление или удаление указанной строки.

3. Как внести изменения в уже сделанный коммит?

`$ git commit --amend`

Отмена может потребоваться, если вы сделали коммит слишком рано, например, забыв добавить какие-то файлы или комментарий к коммиту. Если вы хотите переделать коммит — внесите необходимые изменения, добавьте их в индекс и сделайте коммит ещё раз, указав параметр `--amend`.

Например, если вы сделали коммит и поняли, что забыли проиндексировать изменения в файле, который хотели добавить в коммит, то можно сделать следующее:

`$ git commit -m 'Initial commit'`

`$ git add forgotten_file`

`$ git commit --amend`

4. Как отменить индексацию файла в Git?

use `"git reset HEAD ..." to unstage`

(`$ git reset HEAD CONTRIBUTING.md`) пример для исключения из индекса файла CONTRIBUTING.md. Файл CONTRIBUTING.md изменен, но больше не добавлен в индекс.

5. Как отменить изменения в файле?

use `"git checkout -- ..."` to discard changes in working directory

Важно понимать, что `git checkout --` — опасная команда. Все локальные изменения в файле пропадут — Git просто заменит его версией из последнего коммита. Ни в коем случае не используйте эту команду, если вы не уверены, что изменения в файле вам не нужны.

6. Что такое удаленный репозиторий Git?

Удалённые репозитории представляют собой версии вашего проекта, сохранённые в интернете или ещё где-то в сети.

7. Как выполнить просмотр удаленных репозиториях данного локального репозитория?

Для того, чтобы просмотреть список настроенных удалённых репозиториях, вы можете запустить команду `git remote`. Она выведет названия доступных удалённых репозиториях.

8. Как добавить удаленный репозиторий для данного локального репозитория?

Для того, чтобы добавить удалённый репозиторий и присвоить ему имя (shortname), просто выполните команду `git remote add`:

```
$ git remote add pb https://github.com/paulboone/ticgit
```

9. Как выполнить отправку/получение изменений с удаленного репозитория?

```
$ git fetch [remote-name]
```

Данная команда связывается с указанным удалённым проектом и забирает все те данные проекта, которых у вас ещё нет.

```
$ git push origin master
```

Когда вы хотите поделиться своими наработками, вам необходимо отправить их в удалённый репозиторий. Команда для этого действия простая: `git push`. Чтобы отправить вашу ветку master на сервер origin (повторимся, что клонирование обычно настраивает оба этих имени автоматически), вы можете выполнить следующую команду для отправки ваших коммитов:

10. Как выполнить просмотр удаленного репозитория?

Если хотите получить побольше информации об одном из удалённых репозиториях, вы можете использовать команду `git remote show <remote>`

Она выдаёт URL удалённого репозитория, а также информацию об отслеживаемых ветках. Эта команда любезно сообщает вам, что если вы, находясь на ветке master, выполните `git pull`, ветка master с удалённого сервера будет автоматически влита в вашу сразу после получения всех необходимых данных. Она также выдаёт список всех полученных ею ссылок.

11. Каково назначение тэгов Git?

Теги — это ссылки, указывающие на определенные точки в истории Git.

Команда `git tag` обычно используется для захвата некой точки в истории, которая используется для релиза нумерованной версии (например, v1.0.1).

12. Как осуществляется работа с тэгами Git?

Для создания нового тега выполните следующую команду:

```
git tag <tagname>
```

Просмотреть список имеющихся тегов в Git можно очень просто. Достаточно набрать команду

```
git tag
```

Git использует два основных типа тегов: легковесные и аннотированные.

Легковесный тег — это что-то очень похожее на ветку, которая не изменяется — просто указатель на определённый коммит. А вот аннотированные теги хранятся в базе данных Git как полноценные объекты. Они имеют контрольную сумму, содержат имя автора, его e-mail и дату создания, имеют комментарий и могут быть подписаны и проверены с помощью GNU Privacy Guard (GPG).

Создание аннотированного тега в Git выполняется легко. Самый простой способ — это указать - а при выполнении команды `tag` :

```
$ git tag -a v1.4 -m "my version 1.4"
```

13. Самостоятельно изучите назначение флага --prune в командах git fetch и git push. Каково назначение этого флага?

Для создания зеркала необходимо вначале выполнить команду получения всех изменений с репозитория GitHub

```
$ git fetch --prune
```

(git-fetch - Загрузка объектов и ссылок из другого репозитория)

после чего выполнить отправку содержимого локального репозитория в репозиторий GitLab или BitBucket с помощью команды:

```
$ git push --prune git@example.com:/new-location.git
```

```
+refs/remotes/origin/*:refs/heads/* +refs/tags/*:refs/tags/*
```

соответственно, вместо `git@example.com:/new-location.git` необходимо подставить

адрес репозитория, где будет создано зеркало.