

МИНИСТЕРСТВО НАУКИ ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций.

ОТЧЁТ

по лабораторной работе №2

Дисциплина: «Программирование на Python»

Тема: «Исследование возможностей Git для работы с локальными репозиториями»

Выполнил:

студент 2 курса группы ИВТ-б-о-22-1

Бабенко Артём Тимофеевич

Проверил:

Доцент кафедры инфокоммуникаций

Воронкин Р.А

Работа защищена с оценкой: _____

Ставрополь, 2023

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

Ход работы:

```
d:\git\lab2.1>git clone https://github.com/schacon/simplegit-progit
Cloning into 'simplegit-progit'...
remote: Enumerating objects: 13, done.
remote: Total 13 (delta 0), reused 0 (delta 0), pack-reused 13
Receiving objects: 100% (13/13), done.
Resolving deltas: 100% (3/3), done.
```

Рисунок 1. Копирования репозитория из примера

```
d:\git\lab2.1\simplegit-progit>git log
commit ca82a6dff817ec66f44342007202690a93763949 (HEAD -> master, origin/master,
origin/HEAD)
Author: Scott Chacon <schacon@gmail.com>
Date: Mon Mar 17 21:52:11 2008 -0700

    changed the verison number

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test code

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 10:31:28 2008 -0700

    first commit

d:\git\lab2.1\simplegit-progit>
```

Рисунок 2. Команда git log

```
d:\git\lab2.1\simplegit-progit>git log -p -2
commit ca82a6dff817ec66f44342007202690a93763949 (HEAD -> master, origin/master,
origin/HEAD)
Author: Scott Chacon <schacon@gmail.com>
Date: Mon Mar 17 21:52:11 2008 -0700

    changed the verison number

diff --git a/Rakefile b/Rakefile
index a874b73..8f94139 100644
--- a/Rakefile
+++ b/Rakefile
@@ -5,7 +5,7 @@ require 'rake/gempackagetask'
 spec = Gem::Specification.new do |s|
   s.platform = Gem::Platform::RUBY
   s.name = "simplegit"
-  s.version = "0.1.0"
+  s.version = "0.1.1"
   s.author = "Scott Chacon"
   s.email = "schacon@gmail.com"
   s.summary = "A simple gem for using Git in Ruby code."

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 16:40:33 2008 -0700
```

Рисунок 3. Команда git log с аргументами

```
d:\git\lab2.1\simplegit-progit>git log --stat
commit ca82a6dff817ec66f44342007202690a93763949 <HEAD -> master, origin/master, origin/HEAD>
Author: Scott Chacon <schacon@gmail.com>
Date: Mon Mar 17 21:52:11 2008 -0700

    changed the verison number

Rakefile | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test code

lib/simplegit.rb | 5 -----
1 file changed, 5 deletions(-)

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 10:31:28 2008 -0700

    first commit
```

Рисунок 4. Команда git log --stat

```
d:\git\lab2.1\simplegit-progit>git log --pretty=oneline
ca82a6dff817ec66f44342007202690a93763949 <HEAD -> master, origin/master, origin/HEAD> changed the verison number
085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7 removed unnecessary test code
a11bef06a3f659402fe7563abf99ad00de2209e6 first commit
```

Рисунок 5. Команда git log --pretty=oneline

```
d:\git\lab2.1\simplegit-progit>git log --pretty=format:"%h - %an, %ar : %s"
ca82a6d - Scott Chacon, 16 years ago : changed the verison number
085bb3b - Scott Chacon, 16 years ago : removed unnecessary test code
a11bef0 - Scott Chacon, 16 years ago : first commit
```

Рисунок 6. Команда git log --pretty=format:"%h - %an, %ar : %s"

```
d:\git\lab2.1\simplegit-progit>git log --pretty=format:"%h %s" --graph
* ca82a6d changed the verison number
* 085bb3b removed unnecessary test code
* a11bef0 first commit
```

Рисунок 7. Команда git log --pretty=format:"%h %s" --graph

```
d:\git\lab2.1\simplegit-progit>git log --since=2.weeks
```

Рисунок 8. Команда git log --since=2.weeks

```
d:\git\lab2.1\simplegit-progit>git remote
origin
```

Рисунок 9. Команда git remote

```
d:\git\lab2.1\simplegit-progit>git fetch origin
```

Рисунок 10. Команда git fetch origin

```
d:\git\lab2.1\simplegit-progit>git remote show origin
* remote origin
Fetch URL: https://github.com/schacon/simplegit-progit
Push URL: https://github.com/schacon/simplegit-progit
HEAD branch: master
Remote branch:
  master tracked
Local branch configured for 'git pull':
  master merges with remote master
Local ref configured for 'git push':
  master pushes to master (up to date)
```

Рисунок 11. Команда git remote show origin

```
d:\git\lab2.1\simplegit-progit>git tag -a v1.4 -m "my version 1.4"
d:\git\lab2.1\simplegit-progit>git tag
v1.4
```

Рисунок 12. Команда git tag

```
d:\git\lab2.1\simplegit-progit>git show v1.4
tag v1.4
Tagger: IngaBesedina <inga.besedina@mail.ru>
Date: Mon Oct 9 12:16:33 2023 +0300

my version 1.4

commit ca82a6dff817ec66f44342007202690a93763949 <HEAD -> master, tag: v1.4, origin/master, origin/HEAD>
Author: Scott Chacon <schacon@gmail.com>
Date: Mon Mar 17 21:52:11 2008 -0700

    changed the verison number

diff --git a/Rakefile b/Rakefile
index a874b73..8f94139 100644
--- a/Rakefile
+++ b/Rakefile
@@ -5,7 +5,7 @@ require 'rake/gempackagetask'
 spec = Gem::Specification.new do |s|
   s.platform = Gem::Platform::RUBY
   s.name = "simplegit"
-  s.version = "0.1.0"
+  s.version = "0.1.1"
   s.author = "Scott Chacon"
```

Рисунок 13. Команда git show

```
d:\git\lab2.1\simplegit-progit>git push origin v1.4
remote: Permission to schacon/simplegit-progit.git denied to weilli.
fatal: unable to access 'https://github.com/schacon/simplegit-progit/': The requested URL returned error: 403
```

Рисунок 14. Команда git push origin v1.4

```
d:\git\lab2.1\simplegit-progit>git tag -d v1.4
Deleted tag 'v1.4' (was 54424e0)
```

Рисунок 15. Команда git tag -d v1.4

Конец проработки примера

Начало работы над индивидуальным заданием

```
d:\Gitlab\lab2>git log
commit 196ab702126f8212bd207a7892a8266852c39c1e (HEAD -> main)
Author: Nakakapag <lorianss@mail.ru>
Date: Tue Nov 7 23:23:56 2023 +0300

    add

commit ec0243f9a7cd1a7b4d1e9e5b6d22a3019c30457f
Author: Nakakapag <lorianss@mail.ru>
Date: Tue Nov 7 23:17:40 2023 +0300

    add

commit 80f48102d5a2e4c560b34dd2d5de0eb4358b1ff5 (tag: v1.1, origin/main, origin/HEAD)
Author: Nakakapag <lorianss@mail.ru>
Date: Wed Oct 11 01:15:09 2023 +0300

    code end

commit 7efe8e2988889d1ff1fd718e34ebcf467b76f3bb
Author: Nakakapag <lorianss@mail.ru>
Date: Wed Oct 11 01:14:13 2023 +0300

    code half

commit 8a163b2149e71be63b253d6757431055e1379345
Author: Nakakapag <lorianss@mail.ru>
Date: Wed Oct 11 01:13:07 2023 +0300
```

Рисунок 16. Команда git log

```
d:\Gitlab\lab2>git log --pretty=oneline
196ab702126f8212bd207a7892a8266852c39c1e (HEAD -> main, tag: 1.2) add
ec0243f9a7cd1a7b4d1e9e5b6d22a3019c30457f add
80f48102d5a2e4c560b34dd2d5de0eb4358b1ff5 (tag: v1.1, origin/main, origin/HEAD) code end
7efe8e2988889d1ff1fd718e34ebcf467b76f3bb code half
8a163b2149e71be63b253d6757431055e1379345 start programm code
43a9f827d405c8dc330bbd2be197c8a3df702d99 Add code derectory
4efb3b42c04537be7c86a5324d863d2aa01b5b0c Add info at README
72a136389d41846c956f7ba789b41af23493baa2 Initial commit
```

Рисунок 17. Команда git log —pretty=oneline

```
d:\Gitlab\lab2>git tag
1.2
v1.0
v1.1
```

Рисунок 18. Команда git tag

```
d:\Gitlab\lab2>git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 6 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 121.91 KiB | 10.16 MiB/s, done.
Total 8 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/lorianss/lab2.git
    80f4810..196ab70  main -> main
```

Рисунок 19. Команда push

```

d:\Gitlab\lab2>git show HEAD
commit 196ab702126f8212bd207a7892a8266852c39c1e (HEAD -> main, tag: 1.2, origin/main, origin/HEAD)
Author: Nakakapag <lorianss@mail.ru>
Date: Tue Nov 7 23:23:56 2023 +0300

    add

diff --git a/doc/Lab 2.docx b/doc/Lab 2.docx
index f031831..8b14867 100644
--- a/doc/Lab 2.docx
+++ b/doc/Lab 2.docx
@@ -23,7 +23,7 @@

```

Рисунок 20. Команда git show HEAD

```

d:\Gitlab\lab2>git status
On branch main
Your branch is up to date with 'origin/main'.

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:   code/code.txt
        modified:   doc/Lab 2.docx

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

d:\Gitlab\lab2>git restore code/code.txt

d:\Gitlab\lab2>git status
On branch main
Your branch is up to date with 'origin/main'.

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:   doc/Lab 2.docx

```

Рисунок 20. Откат изменений командой git restore

```

d:\Gitlab\lab2>git reset --hard HEAD~1
HEAD is now at 196ab70 add

```

Рисунок 21. Откат сохранённых изменений командой git reset --hard HEAD

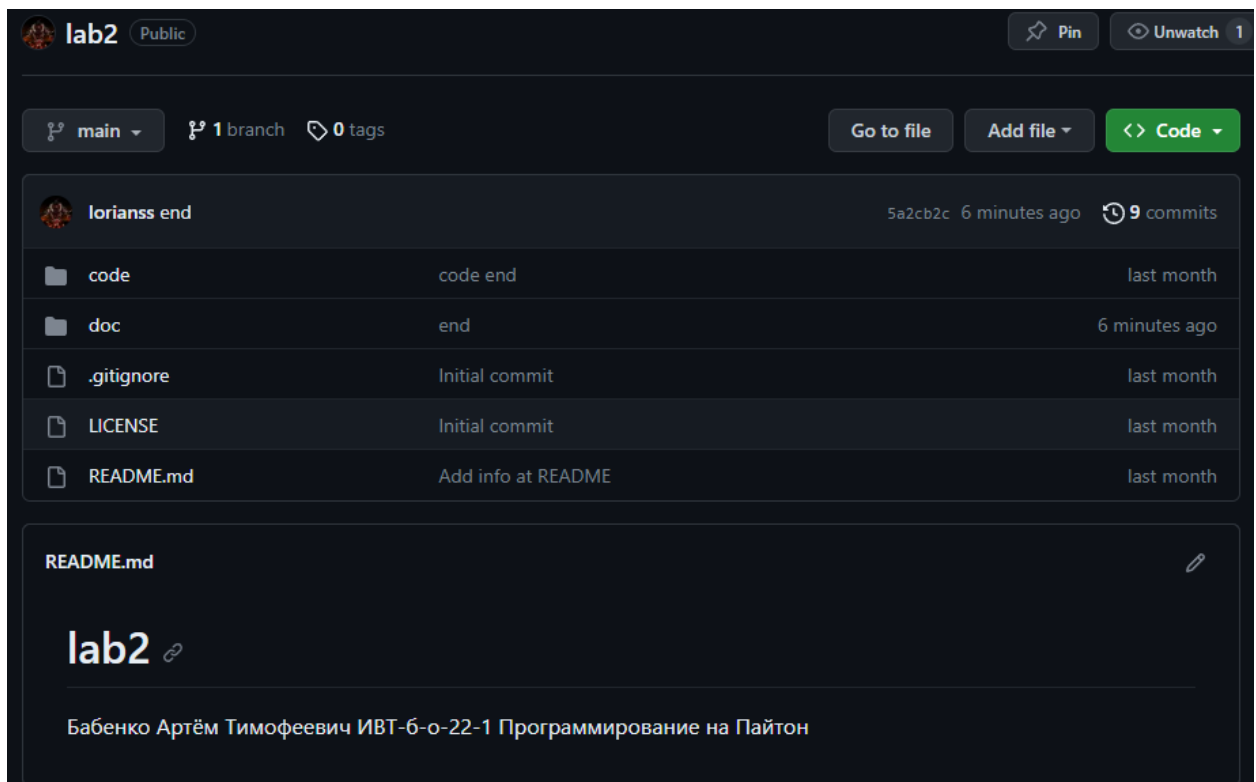


Рисунок 22. Окно браузера после внесение изменений.

Вывод: успешно были исследованы базовые возможности системы контроля версий Git для работы с локальными репозиториями.

Контрольные вопросы:

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

Команда `git log` позволяет видеть историю коммитов. Вместе с этой командой можно использовать аргумент `-p` или `--patch` который показывает разницу (выводит патч), внесенную в каждый коммит. если вы хотите увидеть сокращенную статистику для каждого коммита, вы можете использовать опцию `--stat`. 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. Наиболее интересной опцией является `format`, которая позволяет указать формат для вывода информации.

2. Как ограничить вывод при просмотре истории коммитов?

Вместе с командой `git log` следует использовать параметр `-n` где `n` это число необходимых для вывода записей

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`

`-(n)`

Показывает только последние `n` коммитов.

`--since`, `--after`

Показывает только те коммиты, которые были сделаны после указанной даты.

`--until`, `--before`

Показывает только те коммиты, которые были сделаны до указанной даты.

`--author`

Показывает только те коммиты, в которых запись `author` совпадает с указанной строкой.

`--committer`

Показывает только те коммиты, в которых запись `committer` совпадает с указанной строкой.

`--grep`

Показывает только коммиты, сообщение которых содержит указанную строку.

`-s`

Показывает только коммиты, в которых изменение в коде повлекло за собой добавление или удаление указанной строки.

3. Как внести изменения в уже сделанный коммит?

Если вы хотите переделать коммит — внесите необходимые изменения, добавьте их в индекс и сделайте коммит ещё раз, указав параметр `--amend` :

`git commit --amend`

4. Как отменить индексацию файла в Git?

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

5. Как отменить изменения в файле?

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

6. Что такое удаленный репозиторий Git?

Удалённые репозитории представляют собой версии вашего проекта, сохранённые в интернете или ещё где-то в сети. У вас может быть несколько удалённых репозиториях, каждый из которых может быть доступен для чтения или для чтения-записи. Взаимодействие с другими пользователями предполагает управление удалёнными репозиториями, а также отправку и получение данных из них.

7. Как выполнить просмотр удаленных репозиториях данного локального репозитория?

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

8. Как добавить удаленный репозиторий для данного локального репозитория?


```
git remote add :<shortname> <url>:
```

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

для получения данных из удалённых проектов, следует выполнить:

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

вы можете выполнить следующую команду для отправки ваших коммитов:

```
git push origin master
```

10. Как выполнить просмотр удаленного репозитория?

```
git remote show <remote>
```

11. Каково назначение тэгов Git?

Как и большинство СКВ, Git имеет возможность пометить определённые моменты в истории как важные. Как правило, эта функциональность используется для отметки моментов выпуска версий (v1.0, и т. п.). Такие пометки в Git называются тегами.

12. Как осуществляется работа с тэгами Git?

Можно просмотреть список имеющихся тегов в Git, Достаточно набрать команду `git tag`

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.

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"
```

По умолчанию, команда `git push` не отправляет теги на удалённые сервера. После создания теги нужно отправлять явно на удалённый сервер. Процесс аналогичен отправке веток — достаточно выполнить команду `git push origin` .

```
git push origin v1.5
```

Если у вас много тегов, и вам хотелось бы отправить все за один раз, то можно использовать опцию `--tags` для команды `git push` . В таком случае все ваши теги отправятся на удалённый сервер (если только их уже там нет).

```
$ git push origin --tags
```

Для удаления тега в локальном репозитории достаточно выполнить команду `git tag -d` . Например, удалить созданный ранее легковесный тег можно следующим образом:

```
$ git tag -d v1.4-lw
```

Обратите внимание, что при удалении тега не происходит его удаления с внешних серверов. Существует два способа изъятия тега из внешнего репозитория. Первый способ — это выполнить команду `git push`

```
git push origin :refs/tags/v1.4-lw
```

Второй способ убрать тег из внешнего репозитория более интуитивный:

```
$ git push origin --delete
```

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

`git fetch --prune` — лучшая утилита для очистки устаревших веток. Он подключится к общему удаленному репозиторию и получит все ссылки на удаленные ветки. Затем он удалит удаленные ссылки, которые больше не используются в удаленном репозитории.

`git push --prune` уберёт все ветки которые уже удалены на удалённом репозитории, но ещё есть на локальном.