

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования «СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Институт цифрового развития

ОТЧЁТ

по лабораторной работе №1.2

Дисциплина: «Основы кроссплатформенного программирования»

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

Выполнил: студент 1 курса,
группы ИВТ-б-о-21-1
Богдан Александр Анатольевич

Ставрополь 2022

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

```
D:\WAR\gi\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
```

Рисунок 1. Команда git log

```
D:\WAR\gi\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

    removed unnecessary test code

diff --git a/lib/simplegit.rb b/lib/simplegit.rb
index a0a60ae..47c6340 100644
--- a/lib/simplegit.rb
```

Рисунок 2. Опции git log

```

D:\WAR\gi\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

 README      | 6 +++++
 Rakefile    | 23 ++++++++++++++++++++++
 lib/simplegit.rb | 25 ++++++++++++++++++++++
 3 files changed, 54 insertions(+)

```

Рисунок 3. Опции git log

```

D:\WAR\gi\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 4. Опции git log

```

D:\WAR\gi\simplegit-progit>git log --pretty=format:"%h - %an, %ar : %s"
ca82a6d - Scott Chacon, 14 years ago : changed the verison number
085bb3b - Scott Chacon, 14 years ago : removed unnecessary test code
a11bef0 - Scott Chacon, 14 years ago : first commit

```

Рисунок 5. Опции git log --pretty

```

D:\WAR\gi\simplegit-progit>git log --pretty=format:"%h %s" --graph
* ca82a6d changed the verison number
* 085bb3b removed unnecessary test code
* a11bef0 first commit

```

Рисунок 6. Опции git log --pretty

```

D:\WAR\gi\simplegit-progit>git log --since=2008-01-15
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

```

Рисунок 7. Команда git log --since

```

D:\WAR\gi\simplegit-progit>git remote -v
origin https://github.com/schacon/simplegit-progit (fetch)
origin https://github.com/schacon/simplegit-progit (push)

```

Рисунок 8. Команда git remote

```

D:\WAR\gi\simplegit-progit>git fetch pb
remote: Enumerating objects: 634, done.
remote: Total 634 (delta 0), reused 0 (delta 0), pack-reused 634Receiving objects: 92% (584/634)
Receiving objects: 100% (634/634), 88.93 KiB | 632.00 KiB/s, done.
Resolving deltas: 100% (261/261), done.
From https://github.com/paulboone/ticgit
* [new branch]      master      -> pb/master
* [new branch]      ticgit      -> pb/ticgit

```

Рисунок 9. Команда git fetch

```

D:\WAR\gi\simplegit-progit>git push origin master
remote: Permission to schacon/simplegit-progit.git denied to Alekbs.
fatal: unable to access 'https://github.com/schacon/simplegit-progit/': The requested URL returned error: 403

D:\WAR\gi\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 10. Команда git push

```

D:\WAR\gi\simplegit-progit>git remote rename pb paul

D:\WAR\gi\simplegit-progit>git remote
origin
paul

```

Рисунок 11. Команда git remote rename

```

D:\WAR\gi\simplegit-progit>git show v1.4
tag v1.4
Tagger: Alekbs <aleksandrb436@gmail.com>
Date:   Wed May 18 14:41:20 2022 +0300

my version 1.0

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"
   s.email = "schacon@gmail.com"
   s.summary = "A simple gem for using Git in Ruby code."

```

Рисунок 11. Команда git show

```

D:\WAR\gi\simplegit-progit>git tag -d v1.4
Deleted tag 'v1.4' (was 2b03f06)

```

Рисунок 12. Команда git tag

```

D:\WAR\gi>git clone https://github.com/Alekbs/Lab2
Cloning into 'Lab2'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.

```

Рисунок 13. Клонирование репозитория

```
D:\WAR\gi\Lab2>git add .gitignore

D:\WAR\gi\Lab2>git commit -m "add gitignore"
[main fad667c] add gitignore
 1 file changed, 406 insertions(+)

D:\WAR\gi\Lab2>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), 3.36 KiB | 1.12 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Alekbs/Lab2
   afc45fe..fad667c  main -> main

D:\WAR\gi\Lab2>git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 804 bytes | 2.00 KiB/s, done.
From https://github.com/Alekbs/Lab2
   fad667c..8889195  main       -> origin/main
Updating fad667c..8889195
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

Рисунок 14. Добавление .gitignore

```

D:\WAR\gi\Lab2>git commit -m "1"
[main 023ebe3] 1
4 files changed, 209 insertions(+)
create mode 100644 ConsoleApplication1/ConsoleApplication1.sln
create mode 100644 ConsoleApplication1/ConsoleApplication1/ConsoleApplication1.cpp
create mode 100644 ConsoleApplication1/ConsoleApplication1/ConsoleApplication1.vcxproj
create mode 100644 ConsoleApplication1/ConsoleApplication1/ConsoleApplication1.vcxproj.filters

D:\WAR\gi\Lab2>git add ConsoleApplication1

D:\WAR\gi\Lab2>git commit -m "2"
[main 070b551] 2
1 file changed, 2 insertions(+)

D:\WAR\gi\Lab2>git add ConsoleApplication1

D:\WAR\gi\Lab2>>git commit -m "3"
"commit" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

D:\WAR\gi\Lab2>git commit -m "3"
[main 65cbcb5] 3
1 file changed, 2 insertions(+), 1 deletion(-)

D:\WAR\gi\Lab2>git tag -a v1.0 -m "my version 1.0"

D:\WAR\gi\Lab2>git add ConsoleApplication1

D:\WAR\gi\Lab2>git commit -m "4"
[main e6300a4] 4
1 file changed, 1 insertion(+)

D:\WAR\gi\Lab2>git add ConsoleApplication1

D:\WAR\gi\Lab2>git commit -m "5"
[main dc3a21e] 5
1 file changed, 2 insertions(+), 1 deletion(-)

D:\WAR\gi\Lab2>git tag -a v1.1 -m "my version 1.1"

D:\WAR\gi\Lab2>git add ConsoleApplication1

D:\WAR\gi\Lab2>git commit -m "6"
[main db7959a] 6
1 file changed, 1 insertion(+)

D:\WAR\gi\Lab2>git add ConsoleApplication1

D:\WAR\gi\Lab2>git commit -m "7"
[main 9817d07] 7
1 file changed, 2 insertions(+), 1 deletion(-)

D:\WAR\gi\Lab2>git tag -a v1.1 -m "my version 1.1"
fatal: tag 'v1.1' already exists

D:\WAR\gi\Lab2>git tag -a v1.1 -m "my version 1.1"
fatal: tag 'v1.1' already exists

D:\WAR\gi\Lab2>git tag -a v1.2 -m "my version 1.2"

D:\WAR\gi\Lab2>_

```

Рисунок 15. Добавление коммитов и тегов

```
D:\WAR\gi\Lab2>git log --graph --pretty=oneline --abbrev-commit
* 9817d07 (HEAD -> main, tag: v1.2) 7
* db7959a 6
* dc3a21e (tag: v1.1) 5
* e6300a4 4
* 65cbcb5 (tag: v1.0) 3
* 070b551 2
* 023ebe3 1
* 8889195 (origin/main, origin/HEAD) Create README.md
* fad667c add gitignore
* afc45fe Initial commit
```

Рисунок 16. Просмотр коммитов

```
D:\WAR\gi\Lab2>git show HEAD
commit 9817d071e39c75d266f909527ea4499c646d3ef6 (HEAD -> main, tag: v1.2)
Author: Alekbs <aleksandrb436@gmail.com>
Date: Wed May 18 15:28:24 2022 +0300

7
diff --git a/ConsoleApplication1/ConsoleApplication1/ConsoleApplication1.cpp b/ConsoleApplication1/ConsoleApplication1/ConsoleApplication1.cpp
index 98bd825..2d29a7e 100644
--- a/ConsoleApplication1/ConsoleApplication1/ConsoleApplication1.cpp
+++ b/ConsoleApplication1/ConsoleApplication1/ConsoleApplication1.cpp
@@ -9,7 +9,8 @@ int main()
     cout << "Введите a:";
     cin >> a;
     b = 3 * a * a;
-    cout << "b = " << b;
+    cout << "b = " << b << endl;
+    cout << "a * b =" << a * b;
 }
```

Рисунок 17. Просмотр последнего коммита

```
D:\WAR\gi\Lab2>git reset --hard
HEAD is now at 9817d07 7
```

Рисунок 18. Возвращение к предыдущей версии

Ответы на вопросы:

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

После того, как вы создали несколько коммитов или же клонировали репозиторий с уже существующей историей коммитов, вероятно Вам понадобится возможность посмотреть что было сделано — историю коммитов. Одним из основных и наиболее мощных инструментов для этого является команда `git log`.

Команда `git log` имеет очень большое количество опций для поиска коммитов по разным критериям. Рассмотрим наиболее популярные из них.

Одним из самых полезных аргументов является `-p` или `--patch` , который показывает разницу (выводит патч), внесенную в каждый коммит. Так же вы можете ограничить количество записей в выводе команды.

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?

В дополнение к опциям форматирования вывода, команда `git log` принимает несколько опций для ограничения вывода — опций, с помощью которых можно увидеть определенное подмножество коммитов. Одна из таких опций — это опция `-2` , которая показывает только последние два коммита. В действительности вы можете использовать `-<n>` , где `n` — это любое натуральное число и представляет собой `n` последних коммитов. На практике вы не будете часто использовать эту опцию, потому что Git по умолчанию использует постраничный вывод и вы будете видеть только одну страницу за раз.

Однако, опции для ограничения вывода по времени, такие как `--since` и `--until` , являются очень удобными.

Опция `--author` дает возможность фильтровать по автору коммита, а опция `--grep` искать по ключевым словам в сообщении коммита.

3. Как внести изменения в уже сделанный коммит?

Помните, все что попало в коммит почти всегда Git может восстановить. Можно восстановить даже коммиты из веток, которые были удалены, или коммиты, перезаписанные параметром `--amend`. Но всё, что не было включено в коммит и потеряно — скорее всего, потеряно навсегда.

4. Как отменить индексацию файла в Git?

Использовать `git reset HEAD <file>...` для исключения из индекса.

5. Как отменить изменения в файле?

Использовать `"git reset HEAD <file>..."` для возвращения к версии из последнего коммита.

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. Если вы клонировали репозиторий, то увидите как минимум `origin` — имя по умолчанию, которое Git даёт серверу, с которого производилось клонирование.

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 <shortname> <url>`.

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

Для получения данных из удалённых проектов, следует выполнить `$ git fetch [remote-name]`.

Когда вы хотите поделиться своими наработками, вам необходимо отправить их в удалённый репозиторий. Команда для этого действия простая: `git push <remote-name> <branch-name>` .

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.

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?

Просмотреть список имеющихся тегов в Git можно очень просто. Достаточно набрать команду `git tag` (параметры `-l` и `--list` опциональны).

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

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 <tagname>`.

Для удаления тега в локальном репозитории достаточно выполнить команду `git tag -d <tagname>` .

Если вы хотите получить версии файлов, на которые указывает тег, то вы можете сделать `git checkout` для тега. Однако, это переведёт репозиторий в состояние «detached HEAD», которое имеет ряд неприятных побочных эффектов.

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

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