

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Основы кроссплатформенного программирования**

**Отчет по лабораторной работе №2**

Исследование  
основных возможностей Git и GitHub

Выполнил студент группы

ИВТ-б-о-21-1

Харченко Богдан Романович

« » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил доцент

Кафедры инфокоммуникаций, старший  
преподаватель

Воронкин Р.А.

\_\_\_\_\_  
(подпись)

Ставрополь 2022

**Тема:** исследование возможностей Git для работы с локальными репозиториями.

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

### Выполнение работы.

```
D:\Git\Laba_2>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

D:\Git\Laba_2>
```

Рисунок 1. Команда git status

```
D:\Git\Laba_2>git log
commit f18f116cb6bfa7662ba5a02af43504fdc60399b5 (HEAD -> main)
Author: Konstellation <harchenkobo2003@gmail.com>
Date: Tue Mar 29 15:31:37 2022 +0300

    Классный код

commit 3e40c65407427c39a0072bc318b1d706c939a628
Author: Konstellation <harchenkobo2003@gmail.com>
Date: Tue Mar 29 15:30:57 2022 +0300

    Классный код

commit fc04d82db26610e58960936d993a96142a373f19
Author: Konstellation <harchenkobo2003@gmail.com>
Date: Tue Mar 29 15:28:39 2022 +0300

    Редактировал код еще раз

commit 6c2b0ada15d5b1c036f7de97e59892fc70ee2402
Author: Konstellation <harchenkobo2003@gmail.com>
Date: Tue Mar 29 15:27:38 2022 +0300

    Редактировал код еще раз

commit 461ba1ac504a725708170089a3adb043d8123cc2
Author: Konstellation <harchenkobo2003@gmail.com>
Date: Tue Mar 29 15:26:02 2022 +0300
```

Рисунок 2. Просмотрели коммиты

```

D:\Git\Laba_2>git log -p -2
commit f18f116cb6bfa7662ba5a02af43504fdc60399b5 (HEAD -> main)
Author: Konstellation <harchenkobo2003@gmail.com>
Date: Tue Mar 29 15:31:37 2022 +0300

    Классный код

diff --git a/Laba2/Laba2.cpp b/Laba2/Laba2.cpp
index 8af73e7..8a4ffa1 100644
--- a/Laba2/Laba2.cpp
+++ b/Laba2/Laba2.cpp
@@ -8,4 +8,5 @@ int main()
     cout << 16 + 24 << endl;
     cout << "Hello, Vladimir\n";
     cout << "Last commits\n";
+    return 0;
 }

commit 3e40c65407427c39a0072bc318b1d706c939a628
Author: Konstellation <harchenkobo2003@gmail.com>
Date: Tue Mar 29 15:30:57 2022 +0300

    Классный код

diff --git a/Laba2/Laba2.cpp b/Laba2/Laba2.cpp

```

Рисунок 3. Просмотрели коммиты с фильтром

```

D:\Git\Laba_2>git log --stat
commit f18f116cb6bfa7662ba5a02af43504fdc60399b5 (HEAD -> main)
Author: Konstellation <harchenkobo2003@gmail.com>
Date: Tue Mar 29 15:31:37 2022 +0300

    Классный код

Laba2/Laba2.cpp | 1 +
1 file changed, 1 insertion(+)

commit 3e40c65407427c39a0072bc318b1d706c939a628
Author: Konstellation <harchenkobo2003@gmail.com>
Date: Tue Mar 29 15:30:57 2022 +0300

    Классный код

Laba2/Laba2.cpp | 1 +
1 file changed, 1 insertion(+)

commit fc04d82db26610e58960936d993a96142a373f19
Author: Konstellation <harchenkobo2003@gmail.com>
Date: Tue Mar 29 15:28:39 2022 +0300

    Редактировал код еще раз

```

Рисунок 4. Просмотрели коммиты с сокращенной статистикой

```
D:\Git\Laba_2>git log --pretty=oneline
f18f116cb6bfa7662ba5a02af43504fdc60399b5 (HEAD -> main) Классный код
3e40c65407427c39a0072bc318b1d706c939a628 Классный код
fc04d82db26610e58960936d993a96142a373f19 Редактировал код еще раз
6c2b0ada15d5b1c036f7de97e59892fc70ee2402 Редактировал код еще раз
461ba1ac504a725708170089a3adb043d8123cc2 Редактировал код еще раз
b67d67bbafeff8c26e3b05387e96685f8eb421c2 Изменил код
0d1d8d2992231fd31d72c3973366418d2fb702c6 Изменил код
95b88b9f96090420122c19775b06cd244330feba Изменил gitignore
a95ca2a872e0ceabf6da1d4576004cb0c7516469 Модифицировал gitignore
8bcd9c6ebc2e0646577f4c335aa152c93c61e31 Оформил README
e8a8f127c9a655cabb48ce8fd4d717aba298a719 (origin/main, origin/HEAD) Update README.md
82f50d5f0059a4fc8609bb08a4b33f8220fbc5a8 Initial commit
```

Рисунок 5. Просмотрели коммиты с определенным форматом

```
D:\Git\Laba_2>git log --pretty=format:"%h - %an, %ar : %s"
f18f116 - Konstellation, 10 minutes ago : Классный код
3e40c65 - Konstellation, 11 minutes ago : Классный код
fc04d82 - Konstellation, 13 minutes ago : Редактировал код еще раз
6c2b0ad - Konstellation, 14 minutes ago : Редактировал код еще раз
461ba1a - Konstellation, 16 minutes ago : Редактировал код еще раз
b67d67b - Konstellation, 21 minutes ago : Изменил код
0d1d8d2 - Konstellation, 22 minutes ago : Изменил код
95b88b9 - Konstellation, 24 minutes ago : Изменил gitignore
a95ca2a - Konstellation, 38 minutes ago : Модифицировал gitignore
8bcd9c6 - Konstellation, 40 minutes ago : Оформил README
e8a8f12 - Харченко Богдан, 59 minutes ago : Update README.md
82f50d5 - Харченко Богдан, 60 minutes ago : Initial commit
```

Рисунок 6. Просмотрели коммиты с указанным форматом

```
D:\Git\Laba_2>git log --pretty=format:"%h %s" --graph
* f18f116 Классный код
* 3e40c65 Классный код
* fc04d82 Редактировал код еще раз
* 6c2b0ad Редактировал код еще раз
* 461ba1a Редактировал код еще раз
* b67d67b Изменил код
* 0d1d8d2 Изменил код
* 95b88b9 Изменил gitignore
* a95ca2a Модифицировал gitignore
* 8bcd9c6 Оформил README
* e8a8f12 Update README.md
* 82f50d5 Initial commit
```

Рисунок 7. Просмотрели текущую ветку и историю слияний

```

#include <iostream>
using namespace std;
int main()
{
    cout << "Good morning\n";
    cout << "World!\n";
    cout << 5 + 10<< endl;
    cout << 16 + 24 << endl;
    cout << "Hello, Vladimir\n";
    cout << "Last commits\n";
    return 0;
}

```

Рисунок 8. Код программы

```

D:\Git\Laba_2>git log --graph --pretty=oneline --abbrev-commit
* f18f116 (HEAD -> main) Классный код
* 3e40c65 Классный код
* fc04d82 Редактировал код еще раз
* 6c2b0ad Редактировал код еще раз
* 461ba1a Редактировал код еще раз
* b67d67b Изменил код
* 0d1d8d2 Изменил код
* 95b88b9 Изменил gitignore
* a95ca2a Модифицировал gitignore
* 8bcd9c Оформил README
* e8a8f12 (origin/main, origin/HEAD) Update README.md
* 82f50d5 Initial commit

```

Рисунок 9.Просмотрели историю хранилища

```

D:\Git\Laba_2>git show HEAD
commit f18f116cb6bfa7662ba5a02af43504fdc60399b5 (HEAD -> main)
Author: Konstellation <harchenkobo2003@gmail.com>
Date: Tue Mar 29 15:31:37 2022 +0300

    Классный код

diff --git a/Laba2/Laba2.cpp b/Laba2/Laba2.cpp
index 8af73e7..8a4ffa1 100644
--- a/Laba2/Laba2.cpp
+++ b/Laba2/Laba2.cpp
@@ -8,4 +8,5 @@ int main()
     cout << 16 + 24 << endl;
     cout << "Hello, Vladimir\n";
     cout << "Last commits\n";
+    return 0;
}

```

Рисунок 10. Просмотрели последний коммит

```
D:\Git\Laba_2>git show HEAD~1
commit 3e40c65407427c39a0072bc318b1d706c939a628
Author: Konstellation <harchenkobo2003@gmail.com>
Date: Tue Mar 29 15:30:57 2022 +0300

    Классный код

diff --git a/Laba2/Laba2.cpp b/Laba2/Laba2.cpp
index b5a0dff..8af73e7 100644
--- a/Laba2/Laba2.cpp
+++ b/Laba2/Laba2.cpp
@@ -7,4 +7,5 @@ int main()
     cout << 5 + 10<< endl;
     cout << 16 + 24 << endl;
     cout << "Hello, Vladimir\n";
+    cout << "Last commits\n";
 }
```

Рисунок 11. Просмотрели предпоследний коммит

```
D:\Git\Laba_2>git show b67d67b
commit b67d67bbafeff8c26e3b05387e96685f8eb421c2
Author: Konstellation <harchenkobo2003@gmail.com>
Date: Tue Mar 29 15:21:04 2022 +0300

    Изменил код

diff --git a/Laba2/Laba2.cpp b/Laba2/Laba2.cpp
index 43c640a..2895125 100644
--- a/Laba2/Laba2.cpp
+++ b/Laba2/Laba2.cpp
@@ -3,5 +3,6 @@ using namespace std;
 int main()
 {
     cout << "Hi\n";
-    cout << "World!";
+    cout << "World!\n";
+    cout << 5 + 10;
 }
```

Рисунок 12. Просмотрели коммит с указанным хэшем

```

D:\Git\Laba_2>git status
On branch main
Your branch is ahead of 'origin/main' by 10 commits.
  (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:   Laba2/Laba2.cpp

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

D:\Git\Laba_2>git add .

D:\Git\Laba_2>git commit -m "Удалил код"
[main 0db35cc] Удалил код
1 file changed, 1 insertion(+), 12 deletions(-)

```

Рисунок 13. Удалили код

```

D:\Git\Laba_2>git reset --hard HEAD~1
HEAD is now at f18f116 Классный код

```

Рисунок 14. Откат кода к прошлой версии

```

1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      cout << "Good morning\n";
6      cout << "World!\n";
7      cout << 5 + 10<< endl;
8      cout << 16 + 24 << endl;
9      cout << "Hello, Vladimir\n";
10     cout << "Last commits\n";
11     return 0;
12 }
13

```

Рисунок 15. Откатившийся код



Рисунок 16. Результат работы на Git

**Вывод:** команда `git -checkout <FileName>` удаляет изменения произошедшие с файлом в репозитории до коммита.

### Контрольные вопросы

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

Наиболее простой и в то же время мощный инструмент для этого — команда `git log`. По умолчанию, без аргументов, `git log` выводит список коммитов созданных в данном репозитории в обратном хронологическом порядке. То есть самые последние коммиты показываются первыми. Одна из опций, когда вы хотите увидеть сокращенную статистику для каждого коммита, вы можете использовать опцию `-stat`. Вторая опция (одна 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; используйте параметр `-2` для вывода только двух записей (пример команды `git log -p -2`). Третья действительно полезная опция это `--pretty`. Она 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. 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. Для опции `git log --pretty=format` существуют различного рода опции для изменения формата отображения.

## 2. Как ограничить вывод при просмотре истории коммитов?

Для ограничения может использоваться функция `git log`, где `n` — число записей. Также, существуют опции для ограничения вывода по времени, такие как `--since` и `--until`, они являются очень удобными. Например, следующая команда покажет список коммитов, сделанных за последние две недели: `git log --since=2.weeks`. Это команда работает с большим количеством форматов — вы можете указать определенную дату вида `2008-01-15` или же относительную дату, например `2 years 1 day 3 minutes ago`. Также вы можете фильтровать список коммитов по заданным параметрам. Опция `--author` дает возможность фильтровать по автору коммита, а опция `--grep` (показывает только коммиты, сообщение которых содержит указанную строку) искать по ключевым словам в сообщении коммита.<sup>9</sup> Функция `-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?

Внести изменения можно с помощью команды `git commit --amend`. Эта команда берёт индекс и применяет его к последнему коммиту. Если после последнего коммита не было никаких

проиндексированных изменений (например, вы запустили приведённую команду сразу после предыдущего коммита), то состояние проекта будет абсолютно таким же и всё, что мы изменим, это комментарий к коммиту. Для того, чтобы внести необходимые изменения - нам нужно проиндексировать их и выполнить команду `git commit --amend`. `git commit -m 'initial commit'` `git add forgotten_file` `git commit --amend` Эффект от выполнения этой команды такой, как будто мы не выполнили предыдущий коммит, а еще раз выполнили команду `git add` и выполнили коммит.

#### 4. Как отменить индексацию файла в Git?

Например, вы изменили два файла и хотите добавить их в разные коммиты, но случайно выполнили команду `git add *` и добавили в индекс оба. Как исключить из индекса один из них? Команда `git status` напомним вам: Прямо под текстом «Changes to be committed» говорится: используйте `git reset HEAD` для исключения из индекса.

#### 5. Как отменить изменения в файле?

С помощью команды `git checkout --`.

#### 6. Что такое удаленный репозиторий Git?

Удалённый репозиторий это своего рода наше облако, в которое мы сохраняем те или иные изменения в нашей программе/коде/файлах.

#### 7. Как выполнить просмотр удаленных репозитория данного локального репозитория?

Для того, чтобы просмотреть список настроенных удалённых репозитория, необходимо запустить команду `git remote`. Также можно указать ключ `-v`, чтобы просмотреть адреса для чтения и записи, привязанные к репозиторию. Пример: `git remote -v`

#### 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` .

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

Если необходимо получить изменения, которые есть у Пола, но нету у вас, вы можете выполнить команду `git fetch` . Важно отметить, что команда `git fetch` забирает данные в ваш локальный репозиторий, но не сливает их с какими-либо вашими наработками и не модифицирует то, над чем вы работаете в данный момент. Вам необходимо вручную слить эти данные с вашими, когда вы будете готовы. Если ветка настроена на отслеживание удалённой ветки, то вы можете использовать команду `git pull` чтобы автоматически получить изменения из удалённой ветки и слить их со своей текущей. Выполнение `git pull`, как правило, извлекает (fetch) данные с сервера, с которого вы изначально клонировали, и автоматически пытается слить (merge) их с кодом, над которым вы в данный момент работаете. Чтобы отправить изменения на удалённый репозиторий необходимо отправить их в удалённый репозиторий. Команда для этого действия простая: `git push` .

10. Как выполнить просмотр удаленного репозитория?

Для просмотра удалённого репозитория, можно использовать команду `git remote show` .

11. Каково назначение тэгов Git?

Теги - это ссылки указывающие на определённые версии кода/написанной программы. Они удобны чтобы в случае чего вернуться к нужному моменту. Также при помощи тегов можно помечать важные моменты.

12. Как осуществляется работа с тэгами Git?

Просмотреть наличие тегов можно с помощью команды: `git tag`. А назначить (указать, добавить тег) можно с помощью команды

`git tag -a v1.4`(версия изначальная) `-m "Название"`. С помощью команды `git show` вы можете посмотреть данные тега вместе с коммитом: `git show v1.4`. Отправка тегов, по умолчанию, команда `git push` не отправляет теги на удалённые сервера. После создания тега нужно отправлять явно на удалённый сервер. Процесс аналогичен отправке веток — достаточно выполнить команду `git push origin`. Для отправки всех тегов можно использовать команду `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` Для удаления тега из внешнего репозитория используется команда `git push origin --delete`. Если вы хотите получить версии файлов, на которые указывает тег, то вы можете сделать `git checkout` для тега пример: `git checkout -b version2 v2.0.0`.

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

`Git fetch --prune` команда получения всех изменений с репозитория GitHub. В команде `git push --prune` удаляет удаленные ветки, у которых нет локального аналога. Вывод: исследовал базовые возможности системы контроля версий `git` для работы с локальными репозиториями. Также, благодаря созданию тегов и пункту 7 лабораторной работы после изменения файлов освоил возможность отката к заданной версии.

Вывод: исследовал базовые возможности системы контроля версий `git` для работы с локальными репозиториями. Освоил возможность отката измененных файлов.