

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

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1.2
дисциплины «Основы кроссплатформенного программирования»

Выполнила:
Дудова Мира Сергеевна
1 курс, группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А., доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Исследование возможностей Git для работы с локальными репозиториями

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

Ход работы:

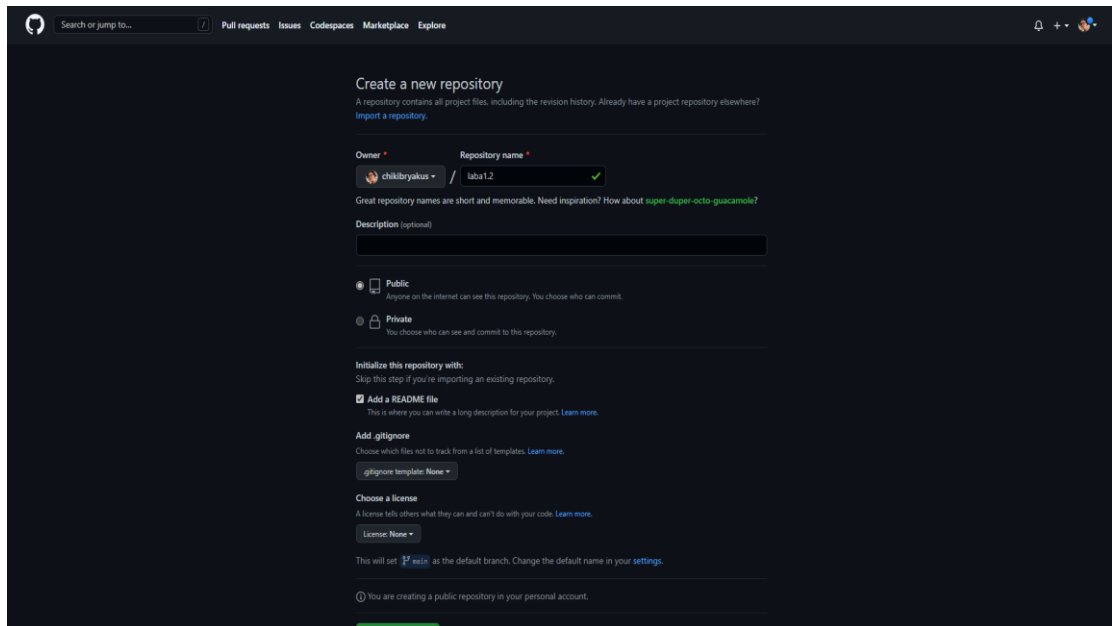


Рисунок 1. Создание репозитория GitHub

```
Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/prorabotat
$ 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.

Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/prorabotat
$ git log
fatal: not a git repository (or any of the parent directories): .git

Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/prorabotat
$ cd C:/Users/Мира/Desktop/prorabotat/simplegit-progit

Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/prorabotat/simplegit-progit (master)
$ 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 version number
```

Рисунок 2. Клонирование репозитория и git log

```

Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/prorabotat/simplegit-progit (master)
$ 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."

```

Рисунок 3. Выведение патча с ограничением в две записи

```

Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/prorabotat/simplegit-progit (master)
$ 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>

```

Рисунок 4. Сокращенная статистика коммитов

```

Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/prorabotat/simplegit-progit (master)
$ 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. Выведение коммитов в одну строку

```

Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/prorabotat/simplegit-progit (master)
$ git log --pretty=format:"%h - %an, %ar : %s"
ca82a6d - Scott Chacon, 15 years ago : changed the verison number
085bb3b - Scott Chacon, 15 years ago : removed unnecessary test code
a11bef0 - Scott Chacon, 15 years ago : first commit

```

Рисунок 6. Форматирование вывода информации

```

Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/prorabotat/simplegit-progit (master)
$ git log --pretty=format:"%h %s" --graph
* ca82a6d changed the verison number
* 085bb3b removed unnecessary test code
* a11bef0 first commit

```

Рисунок 7. Граф, показывающий историю слияний и ветку

```

Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/lab1.1 (main)
$ git commit -m "Add setlocale"
[main 2706e59] Add setlocale
1 file changed, 3 insertions(+), 1 deletion(-)

Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/lab1.1 (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), 322 bytes | 322.00 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/chikibryakus/lab1.1.git
f62ee30..2706e59 main -> main

```

Рисунок 12. Commit 3 и push

```

Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/lab1.1 (main)
$ git commit -m "Add int x and y"
[main 0904d65] Add int x and y
1 file changed, 2 insertions(+), 1 deletion(-)

Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/lab1.1 (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), 290 bytes | 290.00 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/chikibryakus/lab1.1.git
2706e59..0904d65 main -> main

```

Рисунок 13. Commit 4 и push

```
Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/lab1.1 (main)
$ git commit -m "Add code int main"
[main da72a85] Add code int main
1 file changed, 9 insertions(+), 1 deletion(-)

Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/lab1.1 (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), 453 bytes | 453.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/chikibryakus/lab1.1.git
0904d65..da72a85 main -> main
```

Рисунок 14. Commit 5 и push

```
Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/lab1.1 (main)
$ git commit -m "Add int i,c,d"
[main 53d2fce] Add int i,c,d
1 file changed, 2 insertions(+), 1 deletion(-)

Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/lab1.1 (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), 279 bytes | 279.00 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/chikibryakus/lab1.1.git
da72a85..53d2fce main -> main
```

Рисунок 15. Commit 6 и push

```
Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/lab1.1 (main)
$ git commit -m "Add if"
[main aed5bfd] Add if
1 file changed, 15 insertions(+), 1 deletion(-)

Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/lab1.1 (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), 558 bytes | 558.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/chikibryakus/lab1.1.git
53d2fce..aed5bfd main -> main
```

Рисунок 16. Commit 7 и push


```

Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/lab1.1 (main)
$ git commit -m "Add else"
[main 667976d] Add else
1 file changed, 8 insertions(+), 1 deletion(-)

Мира@DESKTOP-34LMQ7R MINGW64 ~/Desktop/lab1.1 (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), 326 bytes | 326.00 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/chikibryakus/lab1.1.git
aed5bfd..667976d main -> main

```

Рисунок 17. Commit 8 и push

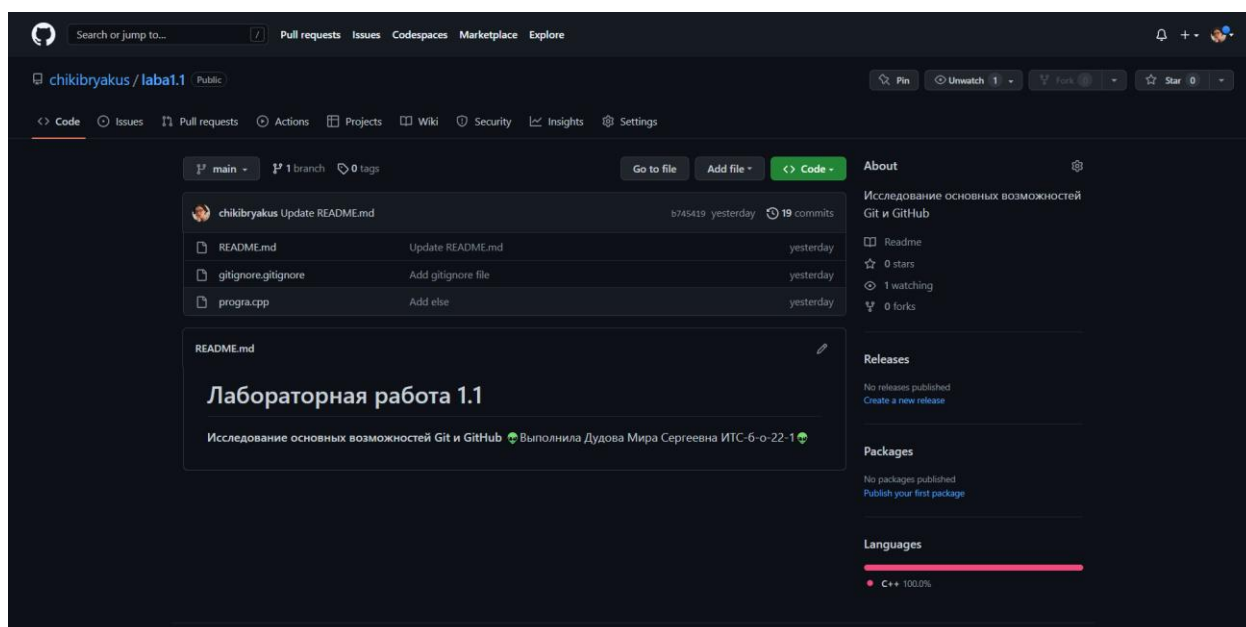


Рисунок 18. Обновленный удаленный репозиторий

Ссылка: <https://github.com/chikibryakus/lab1.1>

Ответы на контрольные вопросы:

1. Что такое СКВ и каково ее назначение?

Система контроля версий (СКВ) — это система, регистрирующая изменения в одном или нескольких файлах этих файлов.

2. В чем недостатки локальных и централизованных СКВ?

Это единая точка отказа, представленная централизованным сервером. Если этот сервер выйдет из строя на час, то в течение этого времени никто не

сможет использовать контроль версий для сохранения изменений, над которыми работает, а также никто не сможет обмениваться этими изменениями с другими разработчиками.

3. К какой СКВ относится Git?

К распределённым системам контроля версий.

4. В чём концептуальное отличие Git от других СКВ?

Git не хранит и не обрабатывает данные таким же способом как другие СКВ.

5. Как обеспечивается целостность хранимых данных в Git?

В Git для всего вычисляется хеш-сумма, и только потом происходит сохранение. В дальнейшем обращение к сохранённым объектам происходит по этой хеш-сумме.

6. В каких состояниях могут находиться файлы в Git? Как связаны эти состояния?

1) Зафиксированный значит, что файл уже сохранён в вашей локальной базе;

2) К изменённым относятся файлы, которые поменялись, но ещё не были зафиксированы;

3) Подготовленные файлы — это изменённые файлы, отмеченные для включения в следующий коммит.

7. Что такое профиль пользователя в GitHub?

Профиль - это наша публичная страница на GitHub, как и в социальных сетях. В нём другие пользователи могут посмотреть ваши работы.

8. Какие бывают репозитории в GitHub?

9. Укажите основные этапы модели работы с GitHub.

1) Регистрация;

2) Создание репозитория;

3) Клонирование репозитория;

4) Добавление новых файлов.

10. Как осуществляется первоначальная настройка Git после установки?

Убедимся, что Git установлен используя команду: `git version`. Перейдём в папку с локальным репозиторием используя команду: `cd /d <Расположения папки на компьютере>`. Свяжем локальный репозиторий и удалённый командами: `git config --global user.name <YOUR_NAME>` `git config --global user.email <EMAIL>`.

11. Опишите этапы создания репозитория в GitHub.

1) В правом верхнем углу, рядом с аватаром есть кнопка с плюсиком, нажимая которую мы переходим к созданию нового репозитория;

2) В результате будет выполнен переход на страницу создания репозитория. Наиболее важными на ней являются следующие поля: Имя репозитория. Описание (Description). Public/private. “Initialize this repository with a README” .gitignore и LICENSE.

12. Какие типы лицензий поддерживаются GitHub при создании репозитория?

Microsoft Reciprocal License, The Code Project Open License (CPOPL), The Common Development and Distribution License (CDDL), The Microsoft Public License (Ms-PL), The Mozilla Public License 1.1 (MPL 1.1), The Common Public License Version 1.0 (CPL), The Eclipse Public License 1.0, The MIT License, The BSD License, The Apache License, Version 2.0, The Creative Commons Attribution-ShareAlike 2.5 License, The zlib/libpng License, A Public Domain dedication, The Creative Commons Attribution 3.0 Unported License, The Creative Commons).

13. Как осуществляется клонирование репозитория GitHub? Зачем нужно клонировать репозиторий?

После создания репозитория его необходимо клонировать на ваш компьютер. Для этого на странице репозитория необходимо найти кнопку Clone или Code и щелкнуть по ней, чтобы отобразить адрес репозитория для клонирования.

Откройте командную строку или терминал и перейдите в каталог, куда вы хотите скопировать хранилище. Затем напишите `git clone` и введите адрес.

14. Как проверить состояние локального репозитория Git?

`git status`

15. Как изменяется состояние локального репозитория Git после выполнения следующих операций: добавления/изменения файла в локальный репозиторий Git; добавления нового/измененного файла под версионный контроль с помощью команды `git add` ; фиксации (коммита) изменений с помощью команды `git commit` и отправки изменений на сервер с помощью команды `git push` ?

Файлы обновятся на репозитории.

16. У Вас имеется репозиторий на GitHub и два рабочих компьютера, с помощью которых Вы можете осуществлять работу над некоторым проектом с использованием этого репозитория. Опишите последовательность команд, с помощью которых оба локальных репозитория, связанных с репозиторием GitHub будут находиться в синхронизированном состоянии.

`git clone.`

`git pull.`

17. GitHub является не единственным сервисом, работающим с Git. Какие сервисы еще Вам известны? Приведите сравнительный анализ одного из таких сервисов с GitHub.

1) GitLab — альтернатива GitHub номер один. GitLab предоставляет не только веб-сервис для совместной работы, но и программное обеспечение с открытым исходным кодом;

2) BitBucket — это служба хостинга репозитория и управления версиями от Atlassian. Она тесно интегрирована с другими инструментами Atlassian — Jira, HipChat и Confluence.

18. Интерфейс командной строки является не единственным и далеко не самым удобным способом работы с Git. Какие Вам известны программные средства с графическим интерфейсом пользователя для работы с Git?

Приведите как реализуются описанные в лабораторной работе операции Git с помощью одного из таких программных средств.

GitHub Desktop это совершенно бесплатное приложение с открытым исходным кодом, разработанное GitHub. С его помощью можно взаимодействовать с GitHub (что и не удивительно), а также с другими платформами (включая Bitbucket и GitLab).

Вывод: исследовала базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub.