

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1.

дисциплина: Математическое моделирование

Студент: Савченков Д.А.

Группа: НПИбд-02-18

МОСКВА

2021 г.

Цели и задачи:

1. Обучиться работе в Git

Описание процесса выполнения работы.

Начал работу в пустом рабочем каталоге с создания пустого каталога с именем hello, затем вошел в него и создал там файл с именем hello.html.

Создал git репозиторий, добавил файл в репозиторий и проверил текущее состояние. (рис. 1)

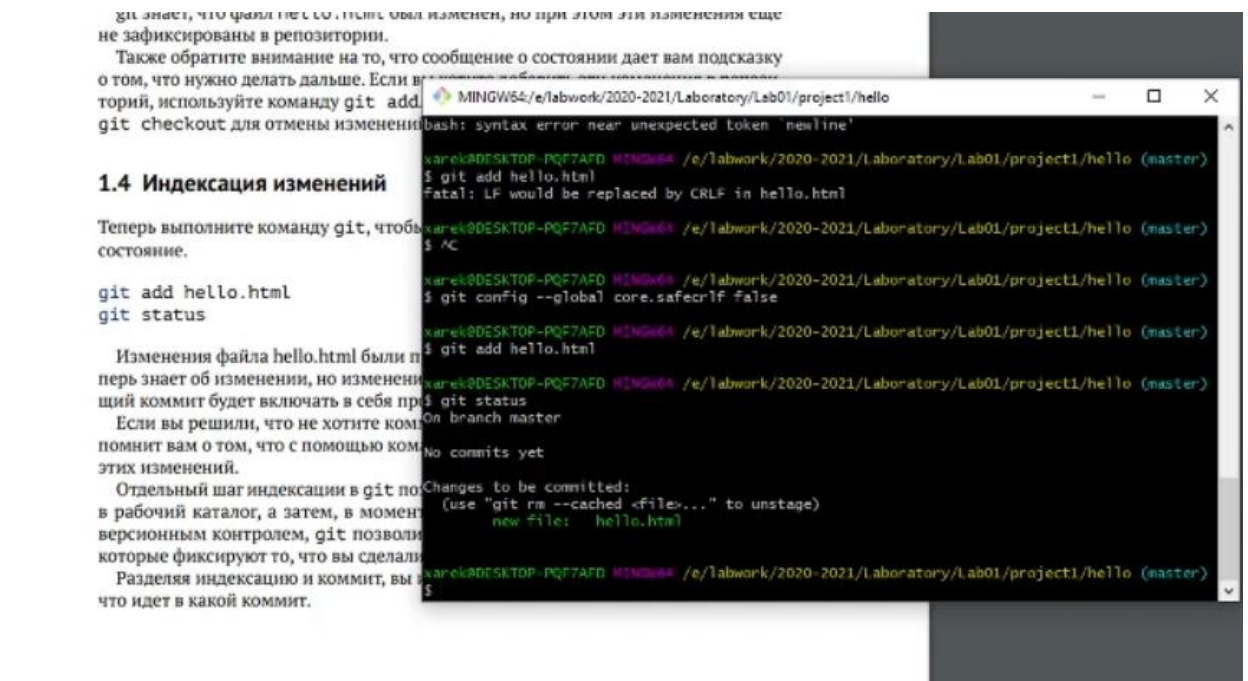


Рис. 1

Добавил HTML теги к нашему приветствию, выполнил команду git, чтобы проиндексировать изменения, проверил состояние. (рис. 2)

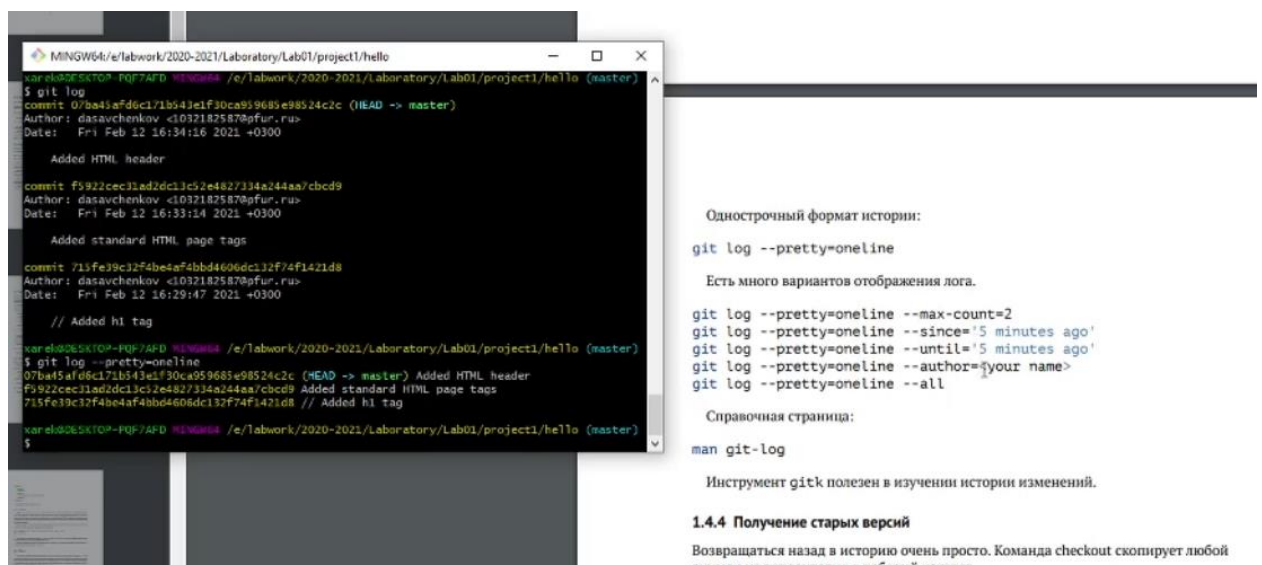


Рис. 2

Проверил истории изменений с начала создания файла. (рис. 3)

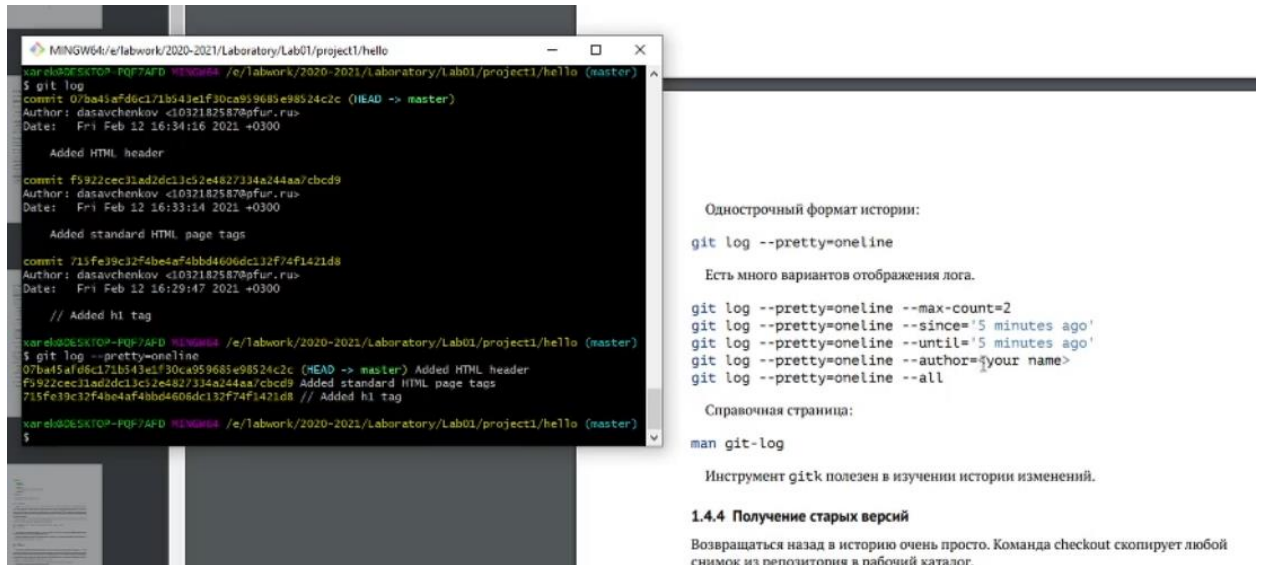


Рис. 3

Проверил старые версии. Получил хэши предыдущих версий, по ним можем перейти в любую старую версию. (рис. 4)

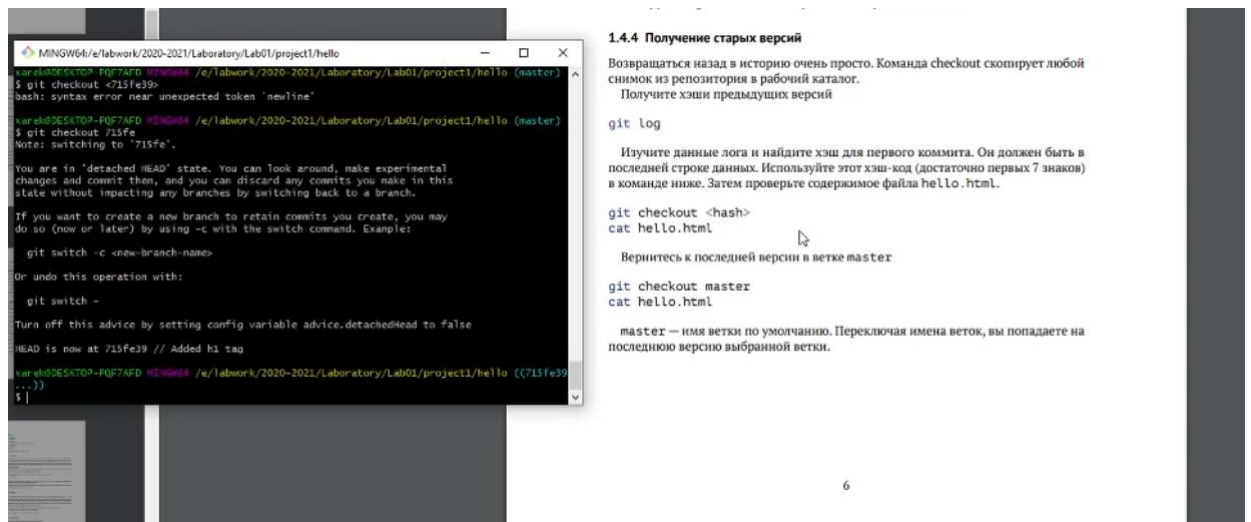


Рис. 4

Создание и отмена коммитов. Изменил файл `hello.html`, добавил ненужную строку и закоммител её, после сделал отмену ненужного коммита. (рис. 5)

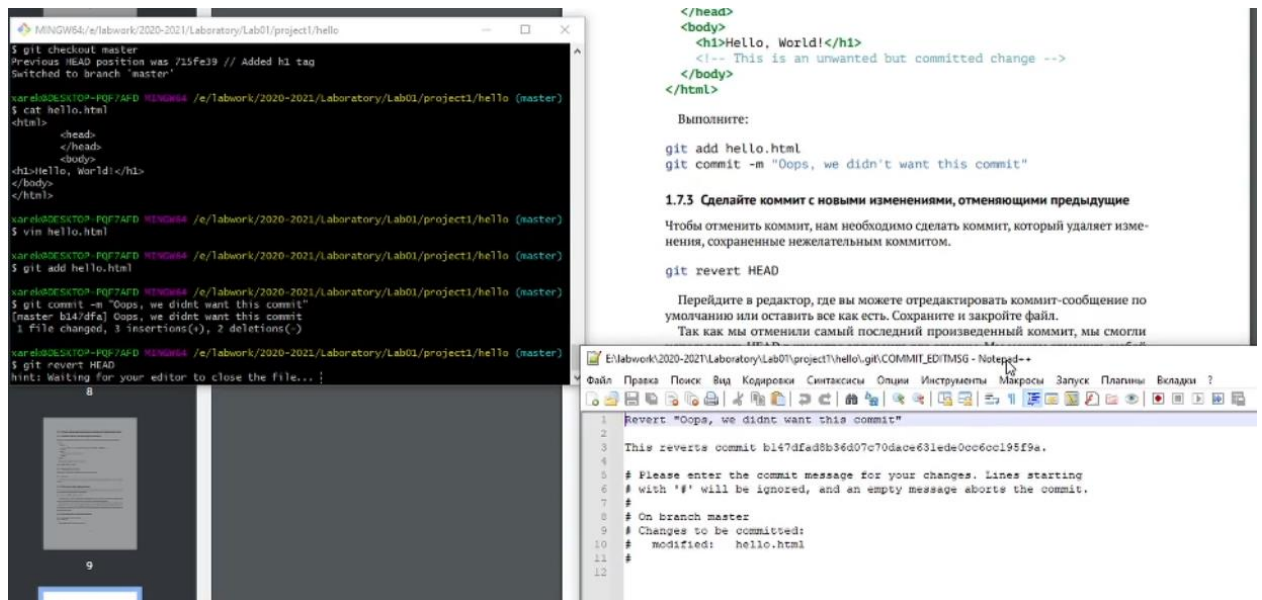


Рис. 5

Затем внес несколько изменений в коммиты. (рис. 6)

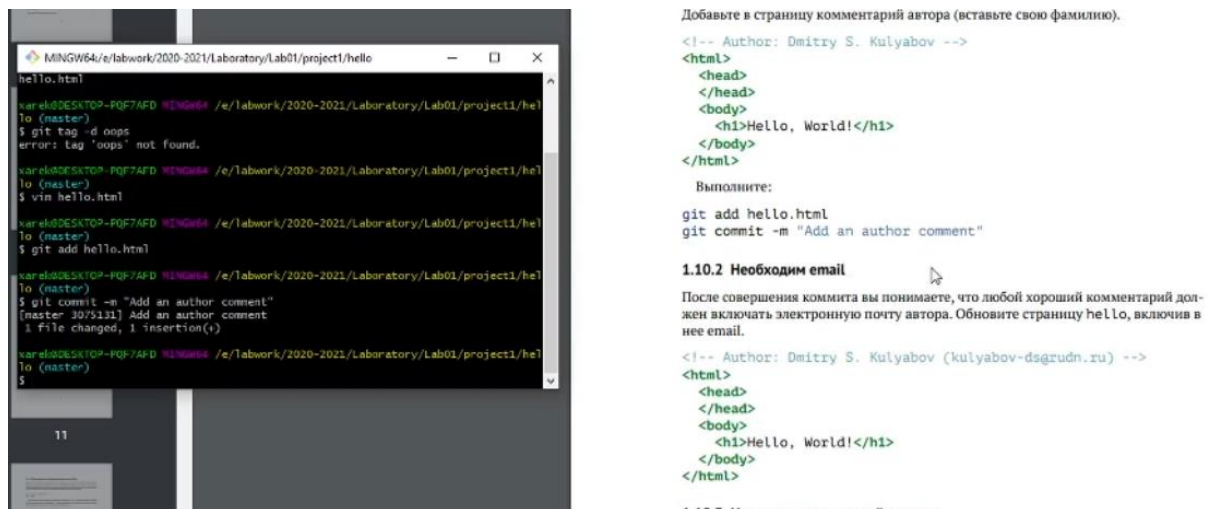
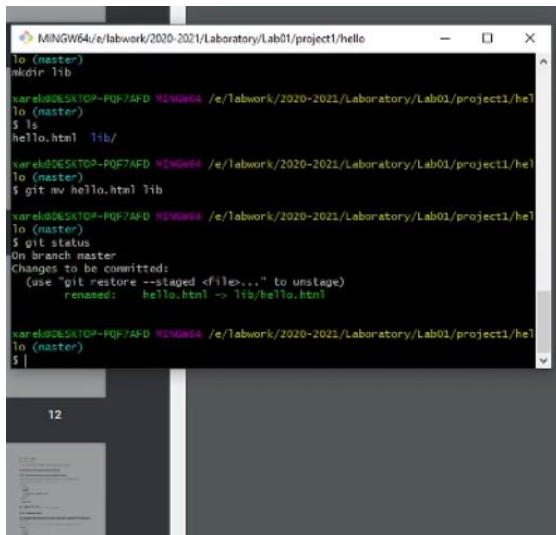


Рис. 6

Перемещение файлов. Перемести файл hello.html в каталог lib, проверил, что действие закончилось успехом. (рис. 7)



Мы можем увидеть, что оригинальный коммит «автор» заменен коммитом «автор/email». Этого же эффекта можно достичь путем сброса последнего коммита в ветке, и повторного коммита новых изменений.

1.11 Перемещение файлов

1.11.1 Переместите файл hello.html в каталог lib

Сейчас мы собираемся создать структуру нашего репозитория. Давайте перенесем страницу в каталог lib.

```
mkdir lib
git mv hello.html lib
git status
```

Перемещая файлы с помощью git mv, мы информируем git о 2 вещах:

- Что файл hello.html был удален.
- Что файл lib/hello.html был создан.
- Оба эти факта сразу же проиндексированы и готовы к коммиту. Команда git status сообщает, что файл был перемещен.

1.12 Второй способ перемещения файлов

Положительной чертой git является то, что вы можете забыть о версионном контроле до того момента, когда вы готовы приступить к коммиту кода. Что бы случилось, если бы мы использовали командную строку операционной системы для перемещения файлов вместо команды git?

Следующий набор команд идентичен нашим последним действиям. Работы здесь побольше, но результат тот же.

Мы могли бы выполнить:

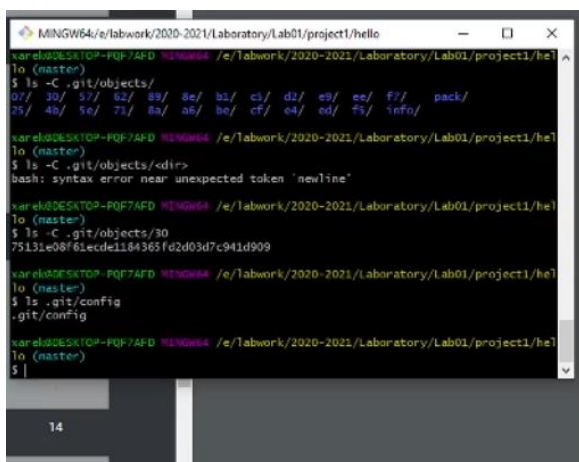
Рис. 7

Структуры. Добавил файл index.html в репозиторий, сделал изменения в файле и коммит. При открытии, увидел, что кусок страницы приветствия находится в маленьком окошке(рамке). (рис. 8)



Рис. 8

Каталог .git. Посмотрел содержание каталога git, проверил базу данных объектов, конфиг файл, ветки и тэги. (рис. 9)



1.14.4 config file

Выполните:

```
cat .git/config
```

Это файл конфигурации, создающийся для каждого конкретного проекта. Записи в этом файле будут перезаписывать записи в файле .gitconfig вашего главного каталога, по крайней мере в рамках этого проекта.

1.14.5 Ветки и теги

Выполните:

```
ls .git/refs
ls .git/refs/heads
ls .git/refs/tags
cat .git/refs/tags/v1
```

Вы должны узнавать файлы в каталоге тегов. Каждый файл соответствует тегу, ранее созданному с помощью команды git tag. Его содержание – это всего лишь хэш коммита, привязанный к тегу.

Каталог heads практически аналогичен, но используется для веток, а не тегов. На данный момент у нас есть только одна ветка, так что все, что вы увидите в этом каталоге – это ветка master.

1.14.6 Файл HEAD

Рис. 9

Работа с объектами git. Посмотрел последний коммит в репозитории, вывел дерево каталогов и наконец, вывел файл hello.html. (рис. 10)

```
MINGW64/e/labwork/2020-2021/Laboratory/Lab01/project1/hello
karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1/hello
$ git cat-file -t 57d92
commit
$ git cat-file -p 57d92
tree 8e8caec7b032c0240dad6b6e5b4574e2ef8322fc
parent 252b27751786dcdb5af83293813074d0b790d78f
author dasavchenkov <10321825878pfur.ru> 1613137808 +0300
committer dasavchenkov <10321825878pfur.ru> 1613137808 +0300
Added index.html
$ git cat-file -p 8e8caec
100644 blob cf3bc0ea8289c0be141b578ca407f1159bd25bc index.html
040000 tree 896dc2d08a5964f23fe9eddc16753fac0406ec lib
$
```

```
git cat-file -p <libhash>

1.15.5 Вывод файла hello.html
Выполните:
git cat-file -p <hellohash>
```

1.15.6 Исследуйте самостоятельно
Исследуйте git репозиторий вручную самостоятельно. Смотрите, удастся ли вам найти оригинальный файл hello.html с самого первого коммита вручную по ссылке SHA1 хэша в последнем коммите.

17

Рис. 10

Создание ветки. Добавил новую ветку style, приукрасил hello world, обновил hello.html, чтобы использовать стили style.css, обновил файл index.html. (рис. 11)

```
MINGW64/e/labwork/2020-2021/Laboratory/Lab01/project1/hello
to (style)
$ cd /e/labwork/2020-2021/Laboratory/Lab01/project1/hello/lib
karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1/hello/lib (style)
$ vim style.css
karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1/hello/lib (style)
$ git add lib/style.css
warning: could not open directory 'lib/lib/': No such file or directory
fatal: pathspec 'lib/style.css' did not match any files
karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1/hello/lib (style)
$ cd /e/labwork/2020-2021/Laboratory/Lab01/project1/hello
karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1/hello (style)
$ git add lib/style.css
karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1/hello (style)
$ add commit -m "Added css stylesheet"
```

```
git status

git checkout -b <имя_ветки> является шорткатом для g
<имя_ветки> за которым идет git checkout <имя_ветки>.
Обратите внимание, что команда git status сообщает о том, что
теперь в ветке «style».
```

1.16.2 Добавьте файл стилей style.css

Выполните:

```
touch lib/style.css
```

Файл lib/style.css:

```
h1 {
  color: red;
}
```

Выполните:

```
git add lib/style.css
git commit -m "Added css stylesheet"
```

Рис. 11

Сделал коммит изменений README.md в ветку master. Добавил файл README.md и сделал коммит. (рис. 12)

```
MINGW64/e/labwork/2020-2021/Laboratory/Lab01/project1/hello
karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1/hello
to (master)
$ echo "This is the Hello World example from git tutorial." > README.md
karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1/hello
to (master)
$ cat README.md
This is the Hello World example from git tutorial.
karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1/hello
to (master)
$ git add README.md
karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1/hello
to (master)
$ git commit -m "Added README"
[master c095a92] Added README
1 file changed, 1 insertion(+)
create mode 100644 README.md
karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1/hello
to (master)
$
```

```
git checkout master
Создайте файл README.md
echo "This is the Hello World example from the git tutorial." > |
```

1.19 Сделайте коммит изменений README .md в ветку master.

Выполните:

```
git add README.md
git commit -m "Added README"
```

1.19.1 Просмотр отличающихся веток

1.19.2 Просмотрите текущие ветки

Теперь у нас в репозитории есть две отличающиеся ветки. Используйте следующую лог-команду для просмотра веток и их отличий.

Выполните:

```
git log --graph --all
```

Добавление опции --graph в git log вызывает построение дерева коммитов с помощью простых ASCII символов. Мы видим обе ветки (style и master), и то, что ветка master является текущей HEAD. Общим предшественником обеих веток является коммит c095a92.

Рис. 12

Клонирование репозитория. Создал клон репозитория hello. (рис. 13)

```
MINGW64/e/labwork/2020-2021/Laboratory/Lab01/project1/cloned_hello
karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1
$ ls
hello/
karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1
$ git clone hello cloned_hello
Cloning into 'cloned_hello'...
done.
karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1
$ ls
cloned_hello/ hello/
karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1
$ cd cloned_hello/
karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1/cloned_hello (master)
$ ls
README.md index.html lib/
karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1/cloned_hello (master)
$ |
```

```
git clone hello cloned_hello
ls
```

В вашем рабочем каталоге теперь должно быть два репозитория: оригинальный репозиторий «hello» и клонированный репозиторий «cloned_hello»

1.28 Просмотр клонированного репозитория

1.28.1 Давайте взглянем на клонированный репозиторий.

Выполните:

```
cd cloned_hello
ls
```

Вы увидите список всех файлов на верхнем уровне оригинального репозитория README.md, index.html и lib.

26

Рис. 13

Удаление ветки. Проверил ветки, доступные в клонированном репозитории. Посмотрел все ветки, внес изменения в оригинальный репозиторий hello. (рис. 14)

```
MINGW64/e/labwork/2020-2021/Laboratory/Lab01/project1/cloned_hello
Remote branches:
  master tracked
  style 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)

karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1/cloned_hello (master)
$ git branch
* master
karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1/cloned_hello (master)
$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/style
karel@DESKTOP-PQF7AFD MINGW64 /e/labwork/2020-2021/Laboratory/Lab01/project1/cloned_hello (master)
$
```

```
cd ../hello
```

Примечание: Сейчас мы находимся в репозитории hello. Внесите следующие изменения в файл README.md: Файл README.md

This is the Hello World example from the git tutorial

Теперь добавьте это изменение и сделайте коммит. Выполните:

```
git add README
git commit -m "Changed README in original repo"
```

Теперь в оригинальном репозитории есть более поздние изменения в клонированной версии. Далее мы извлечем и сольем эти изменения в клонированный репозиторий.

1.31.2 Извлечение изменений

Научитесь извлекать изменения из удаленного репозитория. Выполните:

```
cd ../cloned_hello
```

Рис. 14

Вывод.

Приобрел навыки в использовании Git.