

Отчет по Лабораторной Работе № 1

Работа с git

Нзита Диатезилуа Катенди

Содержание

1	Цель работы	6
2	Задание	7
3	Теоретическое введение	8
4	Выполнение лабораторной работы	9
4.1	Подготовка	9
4.2	Создание проекта	9
4.3	Внесение изменений	10
4.4	История	11
4.5	Получение старых версий	12
4.6	Создание тегов версий	12
4.7	Отмена локальных изменений (до индексации)	13
4.8	Отмена проиндексированных изменений (перед коммитом)	14
4.9	Отмена коммитов	15
4.10	Удаление коммитов из ветки	16
4.11	Удаление тега оорс	16
4.12	Изменение предыдущего коммита	17
4.13	Перемещение файлов	18
4.14	Подробнее о структуре	18
4.15	Git внутри: Каталог .git	19
4.16	Работа непосредственно с объектами git	20
4.17	Создание ветки	21
4.18	Навигация по веткам	21
4.19	Изменения в ветке master	22
4.20	Слияние	23
4.21	Создание конфликта	23
4.22	Разрешение конфликтов	24
4.23	Сброс ветки style	24
4.24	Сброс ветки master	25
4.25	Перебазирование	26
4.26	Слияние в ветку master	26
4.27	Клонирование репозитория	27
4.28	Что такое origin?	27
4.29	Удаленные ветки	28
4.30	Изменение оригинального репозитория	28

4.31	Слияние извлеченных изменений	29
4.32	Добавление ветки наблюдения	30
4.33	Создание чистого репозитория	30
4.34	Отправка и извлечение изменений	31
5	Выводы	33

List of Figures

4.1	Настройка git	9
4.2	Создание репозитория	9
4.3	Внесение изменений в содержимое репозитория	10
4.4	Внесение нескольких изменений в содержимое репозитория	11
4.5	Просмотр истории	11
4.6	Просмотр разных версий репозитория	12
4.7	Создание тегов версий	13
4.8	Переключение по имени тега и просмотр доступных тегов	13
4.9	Отмена локальных изменений (до индексации)	14
4.10	Отмена проиндексированных изменений (перед коммитом)	15
4.11	Отмена коммитов	16
4.12	Удаление коммитов из ветки	16
4.13	Удаление тега оорс	17
4.14	Изменение предыдущего коммита	18
4.15	Перемещение файлов	18
4.16	Добавление нового файла в репозиторий	19
4.17	Добавление нового файла в репозиторий	19
4.18	Каталог .git	20
4.19	Работа непосредственно с объектами git	20
4.20	Поиск оригинального файла hello.html	20
4.21	Создание ветки	21
4.22	Просмотр логов новой ветки	21
4.23	Переключение между ветками	22
4.24	Изменения в ветке master	22
4.25	Просмотр веток	22
4.26	Слияние веток	23
4.27	Создание конфликта	23
4.28	Разрешение конфликта	24
4.29	Поиск коммита перед слиянием	24
4.30	Сброс ветки style	25
4.31	Поиск коммита перед конфликтом	25
4.32	Сброс ветки master	26
4.33	Перебазирование	26
4.34	Слияние style в master	27
4.35	Клонирование репозитория	27
4.36	Просмотр имени по умолчанию удаленного репозитория	28
4.37	Просмтр доступных веток	28

4.38	Изменение оригинального репозитория	29
4.39	Извлечение изменений	29
4.40	Слияние извлеченных изменений	30
4.41	Добавление ветки наблюдения	30
4.42	Создание чистого репозитория	31
4.43	Отправка и извлечение изменений	32

1 Цель работы

Приобретение практических навыков работы с ситемой управления версиями Git.

2 Задание

Провести базовую настройку гит, создать проект и, используя его, изучить основные действия с репозиторием.

3 Теоретическое введение

Git — распределённая система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года; координатор — Дзюн Хамано (**wiki:bash?**).

Работа выполнена с помощью веб-сервиса GitHub (**git:bash?**).

4 Выполнение лабораторной работы

4.1 Подготовка

Сначала настроим `core.autocrlf` с параметрами `true` и `input`, чтобы сделать все переводы строк текстовых файлов в главном репозитории одинаковыми, а затем настроим отображение `unicode`(рис. fig. 4.1).

```
dknzita@dknzita-VirtualBox:~$ git config --global core.autocrlf input
dknzita@dknzita-VirtualBox:~$ git config --global core.safecrlf true
dknzita@dknzita-VirtualBox:~$ git config --global core.quotepath off
dknzita@dknzita-VirtualBox:~$
```

Figure 4.1: Настройка git

4.2 Создание проекта

Создадим пустой каталог `hello`, а в нём файл с именем `hello.html`. Затем создадим `git` репозиторий из этого каталога, выполнив команду `git init`. Добавим файл в репозиторий и проверим статус, который сообщает, что коммитить нечего(рис. fig. 4.2).

```
dknzita@dknzita-VirtualBox:~$ git config --global core.autocrlf input
dknzita@dknzita-VirtualBox:~$ git config --global core.safecrlf true
dknzita@dknzita-VirtualBox:~$ git config --global core.quotepath off
dknzita@dknzita-VirtualBox:~$ mkdir hello
mkdir: cannot create directory 'hello': File exists
dknzita@dknzita-VirtualBox:~$ cd hello
dknzita@dknzita-VirtualBox:~/hello$ touch hello.html
dknzita@dknzita-VirtualBox:~/hello$ echo "Hello, world!" > hello.html
>
```

Figure 4.2: Создание репозитория

4.3 Внесение изменений

Изменим содержимое файла hello.html на:

```
<h1>Hello, World!</h1>
```

Проверив состояние рабочего каталога увидим, что git знает, что файл hello.html был изменен, но при этом эти изменения еще не зафиксированы в репозитории. Теперь проиндексируем изменения и снова посмотрим статус, в нём указано, что изменения пока не записаны в репозиторий. И наконец закоммитим изменения, внеся их в репозиторий и снова посмотрим статус, который теперь показывает, что все изменения внесены в репозиторий (рис. fig. 4.3).

```
dknzita@dknzita-VirtualBox:~/hello$ touch hello.html
dknzita@dknzita-VirtualBox:~/hello$ echo "Hello world!" > hello.html
dknzita@dknzita-VirtualBox:~/hello$ cat hello.html
Hello world!
dknzita@dknzita-VirtualBox:~/hello$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:     git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:     git branch -m <name>
Initialized empty Git repository in /home/dknzita/hello/.git/
dknzita@dknzita-VirtualBox:~/hello$ git add hello.html
dknzita@dknzita-VirtualBox:~/hello$ git commit -m "Initial Commit"
[master (root-commit) 309c3cd] Initial Commit
1 file changed, 1 insertion(+)
create mode 100644 hello.html
dknzita@dknzita-VirtualBox:~/hello$
```

Figure 4.3: Внесение изменений в содержимое репозитория

Изменим страницу «Hello, World», чтобы она содержала стандартные теги

и

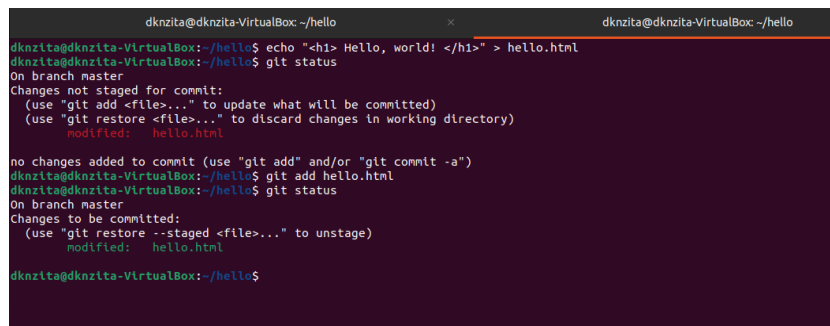
.

```
<html>
  <body>
    <h1>Hello, World!</h1>
  </body>
```

</html>

Теперь добавим это изменение в индекс git и добавим заголовки HTML (секцию

) к странице «Hello, World». Проверив текущий статус увидим, что hello.html указан дважды в состоянии. Первое изменение (добавление стандартных тегов) проиндексировано и готово к коммиту. Второе изменение (добавление заголовков HTML) является непроиндексированным. Произведем коммит проиндексированного изменения, затем проиндексируем оставшееся изменение, посмотрим статус и прокоммитим его(рис. fig. 4.4).



```
dknrita@dknrita-VirtualBox: ~/hello
dknrita@dknrita-VirtualBox:~/hello$ echo "<h1> Hello, world! </h1>" > hello.html
dknrita@dknrita-VirtualBox:~/hello$ git status
On branch master
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:   hello.html

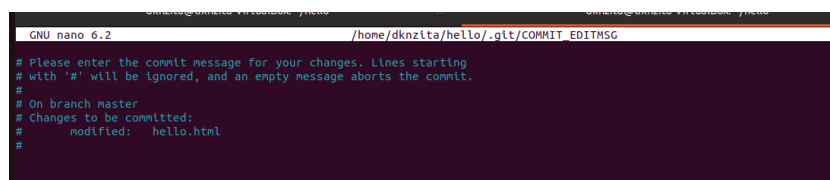
no changes added to commit (use "git add" and/or "git commit -a")
dknrita@dknrita-VirtualBox:~/hello$ git add hello.html
dknrita@dknrita-VirtualBox:~/hello$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

dknrita@dknrita-VirtualBox:~/hello$
```

Figure 4.4: Внесение нескольких изменений в содержимое репозитория

4.4 История

Посмотрим список произведённых изменений в стандартном виде, затем в однострочном, а также с указанием времени и количества(рис. fig. 4.5).



```
GNU nano 6.2 /home/dknrita/hello/.git/COMMIT_EDITMSG
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
#
# Changes to be committed:
#   modified:   hello.html
#
```

Figure 4.5: Просмотр истории

4.5 Получение старых версий

Изучим данные лога и найдем там хэш первого коммита, используя его вернемся к первой версии и посмотрим файл `hello.html`, действительно, увидим первую версию. Затем вернемся к последней версии в ветке `master` и вновь посмотрим на файл(рис. fig. 4.6).

```
dknzita@dknzita-VirtualBox:~/hello$ git commit
Aborting commit due to empty commit message.
dknzita@dknzita-VirtualBox:~/hello$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

dknzita@dknzita-VirtualBox:~/hello$ mcedit hello.html

dknzita@dknzita-VirtualBox:~/hello$ git add hello.html
dknzita@dknzita-VirtualBox:~/hello$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

dknzita@dknzita-VirtualBox:~/hello$
```

Figure 4.6: Просмотр разных версий репозитория

4.6 Создание тегов версий

Назовем текущую версию страницы `hello` первой (`v1`). Создадим тег первой версии и используем его для того чтобы вернуться к предыдущей, которой также присвоим тег(рис. fig. 4.7).

```

dknzita@dknzita-VirtualBox:~/hello$ git commit -m "Added standard HTML tags"
[master b2dfc6a] Added standard HTML tags
1 file changed, 5 insertions(+), 1 deletion(-)
dknzita@dknzita-VirtualBox:~/hello$ git status
On branch master
nothing to commit, working tree clean
dknzita@dknzita-VirtualBox:~/hello$ git add
Nothing specified, nothing added.
hint: Maybe you wanted to say 'git add .' ?
hint: Turn this message off by running
hint: "git config advice.addEmptyPathsSpec false"
dknzita@dknzita-VirtualBox:~/hello$ git status
On branch master
nothing to commit, working tree clean
dknzita@dknzita-VirtualBox:~/hello$ git commit -m "Added HTML header"
On branch master
nothing to commit, working tree clean
dknzita@dknzita-VirtualBox:~/hello$

```

Figure 4.7: Создание тегов версий

Переключимся по тегам между двумя отмеченными версиями. Посмотрим все доступные теги(их два) и посмотрим теки в логе(рис. fig. 4.8).

```

dknzita@dknzita-VirtualBox:~/hello$ git log
commit b2dfc6a1e869e33f2583d980c1a933cf6d667160 (HEAD -> master)
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 12:03:10 2024 +0300

    Added standard HTML tags

commit 309c3cdde59c789445c10d1e1851b4b5fe1051eb
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 11:45:32 2024 +0300

    Initial Commit
dknzita@dknzita-VirtualBox:~/hello$ git log --pretty=oneline
b2dfc6a1e869e33f2583d980c1a933cf6d667160 (HEAD -> master) Added standard HTML tags
309c3cdde59c789445c10d1e1851b4b5fe1051eb Initial Commit
dknzita@dknzita-VirtualBox:~/hello$

```

Figure 4.8: Переключение по имени тега и просмотр доступных тегов

4.7 Отмена локальных изменений (до индексации)

Убедимся, что мы находимся на последнем коммите ветки master и внесем изменение в файл hello.html в виде нежелательного комментария. Затем проверим статус, увидим, что изменения ещё не проиндексированы. Используем команду git checkout для переключения версии файла hello.html в репозитории(рис. fig. 4.9).

```

dknzita@dknzita-VirtualBox:~/hello$ git log
commit b2dfc6a1e869e33f2583d980c1a933cf6d667160 (HEAD -> master)
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 12:03:10 2024 +0300

    Added standard HTML tags

commit 309c3cdd59c789445c10d1e1851b4b5fe1051eb
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 11:45:32 2024 +0300

    Initial Commit

dknzita@dknzita-VirtualBox:~/hello$ git log --pretty=oneline
b2dfc6a1e869e33f2583d980c1a933cf6d667160 (HEAD -> master) Added standard HTML tags
309c3cdd59c789445c10d1e1851b4b5fe1051eb Initial Commit
dknzita@dknzita-VirtualBox:~/hello$ git log --pretty=oneline --max-count=2
fatal: unrecognized argument: --pretty=oneline
dknzita@dknzita-VirtualBox:~/hello$ git log --pretty=oneline --max-count=2
b2dfc6a1e869e33f2583d980c1a933cf6d667160 (HEAD -> master) Added standard HTML tags
309c3cdd59c789445c10d1e1851b4b5fe1051eb Initial Commit
dknzita@dknzita-VirtualBox:~/hello$ git log --pretty=oneline --since='5 minute ago'
fatal: ambiguous argument '5 minute ago': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'
dknzita@dknzita-VirtualBox:~/hello$ git log --pretty=oneline --since='5 minutes ago'
b2dfc6a1e869e33f2583d980c1a933cf6d667160 (HEAD -> master) Added standard HTML tags
dknzita@dknzita-VirtualBox:~/hello$ git log --pretty=oneline --until='5 minutes ago'
309c3cdd59c789445c10d1e1851b4b5fe1051eb Initial Commit
dknzita@dknzita-VirtualBox:~/hello$ git log --pretty=oneline --author=<dknzita>
bash: syntax error near unexpected token `newline'
dknzita@dknzita-VirtualBox:~/hello$ git log --pretty=oneline --author='dknzita'
b2dfc6a1e869e33f2583d980c1a933cf6d667160 (HEAD -> master) Added standard HTML tags
309c3cdd59c789445c10d1e1851b4b5fe1051eb Initial Commit
dknzita@dknzita-VirtualBox:~/hello$ git log --pretty=oneline --all
b2dfc6a1e869e33f2583d980c1a933cf6d667160 (HEAD -> master) Added standard HTML tags
309c3cdd59c789445c10d1e1851b4b5fe1051eb Initial Commit
dknzita@dknzita-VirtualBox:~/hello$

```

Figure 4.9: Отмена локальных изменений (до индексации)

4.8 Отмена проиндексированных изменений (перед коммитом)

Внесем изменение в файл hello.html в виде нежелательного комментария

```

<html>
  <head>
    <!-- This is an unwanted but staged comment -->
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>

```

Проиндексируем это изменение и проверим состояние. Состояние показывает, что изменение было проиндексировано и готово к коммиту. Используем ко-

манду `git reset`, чтобы сбросить буферную зону к HEAD. Это очищает буферную зону от изменений, которые мы только что проиндексировали. И переключимся на последнюю версию коммита, посмотрев статус увидим, что наш каталог опять чист(рис. fig. 4.10).



Figure 4.10: Отмена проиндексированных изменений (перед коммитом)

4.9 Отмена коммитов

Изменим файл `hello.html` на следующий.

```
<html>
  <head>
</head>
  <body>
    <h1>Hello, World!</h1>
    <!-- This is an unwanted but committed change -->
  </body>
</html>
```

Проиндексируем изменения файла и прокоммитим их. Чтобы отменить коммит, нам необходимо сделать коммит, который удаляет изменения, сохранен-

ные нежелательным коммитом. Перейдем в редактор, где изменим нежелательный коммит. Проверим лог. Проверка лога показывает нежелательные и отмененные коммиты в наш репозиторий(рис. fig. 4.11).

```
Author: dknzita <diatezilua.n@gmail.com>
Date:   Mon Apr 22 12:03:10 2024 +0300

    Added standard HTML tags

commit 309c3cdde59c789445c10d1e1851b4b5fe1051eb
Author: dknzita <diatezilua.n@gmail.com>
Date:   Mon Apr 22 11:45:32 2024 +0300

    Initial Commit
dknzita@dknzita-VirtualBox:~/hello$
```

Figure 4.11: Отмена коммитов

4.10 Удаление коммитов из ветки

Удалим последние два коммита с помощью сброса, сначала отметим последний коммит тегом, чтобы его можно было потом найти. Используем команду `git reset`, чтобы вернуться к версии до этих коммитов. Теперь в логе их нет, но если посмотреть логи с опцией `-all` можно всё ещё их увидеть, но метка HEAD находится на нужной нам версии(рис. fig. 4.12).

```
Initial Commit
dknzita@dknzita-VirtualBox:~/hello$ git checkout <hash>
bash: syntax error near unexpected token `newline'
dknzita@dknzita-VirtualBox:~/hello$ cat hello.html
<html>
  <body>
    <h1> Hello, World! </h1>
  </body>
</html>dknzita@dknzita-VirtualBox:~/hello$
```

Figure 4.12: Удаление коммитов из ветки

4.11 Удаление тега oops

Удалим тег `oops` и коммиты, на которые он ссылался, сборщиком мусора. Теперь этот тег не отображается в репозитории(рис. fig. 4.13).


```

dknzita@dknzita-VirtualBox:~/hello$ git tag v1
dknzita@dknzita-VirtualBox:~/hello$ git checkout v1^
Note: switching to 'v1^'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 309c3cd Initial Commit
dknzita@dknzita-VirtualBox:~/hello$ cat hello.html
Hello world!
dknzita@dknzita-VirtualBox:~/hello$ git tag v1-beta
dknzita@dknzita-VirtualBox:~/hello$ git checkout v1
Previous HEAD position was 309c3cd Initial Commit
HEAD is now at b2dfc6a Added standard HTML tags
dknzita@dknzita-VirtualBox:~/hello$ git checkout v1-beta
Previous HEAD position was b2dfc6a Added standard HTML tags
HEAD is now at 309c3cd Initial Commit
dknzita@dknzita-VirtualBox:~/hello$

```

Figure 4.13: Удаление тега oops

4.12 Изменение предыдущего коммита

Добавим в страницу комментарий автора.

```

<!-- Author: Dmitry S. Kulyabov -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>

```

Затем добавим их в репозиторий. Теперь мы хотим добавить в комментарий автора почту, обновиим страницу hello, включив в неё почту. Чтобы у нас остался один коммит, а не два, изменим последний с помощью опции `-amend`, теперь в логах отображается последняя версия коммита(рис. fig. 4.14).

```
dknzita@dknzita-VirtualBox:~/hello$ git tag
v1
v1-beta
dknzita@dknzita-VirtualBox:~/hello$ git log master --all
commit b2dfc6a1e869e33f2583d980c1a933cf6d667160 (tag: v1, master)
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 12:03:10 2024 +0300

    Added standard HTML tags

commit 309c3cdde59c789445c10d1e1851b4b5fe1051eb (HEAD, tag: v1-beta)
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 11:45:32 2024 +0300

    Initial Commit
dknzita@dknzita-VirtualBox:~/hello$
```

Figure 4.14: Изменение предыдущего коммита

4.13 Перемещение файлов

Переместим наш файл в каталог `lib`. Для этого создадим его и используем команду `git mv`, сделаем коммит этого перемещения(рис. fig. 4.15).

```
/home/dknzita/hello/hello.html [----] 7 L: [ 1+ 5 6/ 6] *(117 / 117b) <EOF>
<html>
  <body>
    <!--><h1> Hello, World! </h1>
    <!--><!-- This is a bad comment, we want to revert it -->
  </body>
</html>
```

Figure 4.15: Перемещение файлов

4.14 Подробнее о структуре

Добавим файл `index.html` в наш репозиторий

```
<html>
  <body>
    <iframe src="lib/hello.html" width="200" height="200" />
  </body>
</html>
```

Добавим файл и сделаем коммит(рис. fig. 4.16).

```

dknzita@dknzita-VirtualBox:~/hello$ git status
On branch master
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:   hello.html

no changes added to commit (use "git add" and/or "git commit -a")
dknzita@dknzita-VirtualBox:~/hello$

```

Figure 4.16: Добавление нового файла в репозиторий

Теперь при открытии index.html, увидим кусок страницы hello в маленьком окошке(рис. fig. 4.17).

```

dknzita@dknzita-VirtualBox:~/hello$ git status
On branch master
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:   hello.html

no changes added to commit (use "git add" and/or "git commit -a")
dknzita@dknzita-VirtualBox:~/hello$ git checkout hello.html
Updated 1 path from the index
dknzita@dknzita-VirtualBox:~/hello$ git status
On branch master
nothing to commit, working tree clean
dknzita@dknzita-VirtualBox:~/hello$ cat hello.html
<html>
  <body>
    <h1> Hello, World! </h1>
  </body>
</html>
dknzita@dknzita-VirtualBox:~/hello$

```

Figure 4.17: Добавление нового файла в репозиторий

4.15 Git внутри: Каталог .git

Посмотрим каталог, в котором хранится вся информация git. Затем посмотрим набор каталогов, имена которых состоят из 2 символов. Имена каталогов являются первыми двумя буквами хэша sha1 объекта, хранящегося в git. Посмотрим в один из каталогов с именем из 2 букв. Увидим файлы с именами из 38 символов. Это файлы, содержащие объекты, хранящиеся в git. Посмотрим файл конфигурации, создающийся для каждого конкретного проекта. Затем посмотрим подкаталоги .git/refs/heads и .git/refs/tags, а также содержимое файла v1, в нём хранится хэш коммита, привязанный к тегу. Также посмотрим содержимое файла HEAD, который содержит ссылку на текущую ветку, в данный момент это ветка master(рис. fig. 4.18).

```

</html>dknzita@dknzita-VirtualBox:~/hello$ mcedit hello.html

dknzita@dknzita-VirtualBox:~/hello$ git add hello.html
dknzita@dknzita-VirtualBox:~/hello$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

dknzita@dknzita-VirtualBox:~/hello$ █

```

Figure 4.18: Каталог .git

4.16 Работа непосредственно с объектами git

Найдем последний коммит и выведем его с помощью SHA1 хэша. Затем посмотрим дерево каталогов, ссылка на который идёт в последнем коммите, выведем каталог lib и файл hello.html(рис. fig. 4.19).

```

dknzita@dknzita-VirtualBox:~/hello$ git reset HEAD hello.html
Unstaged changes after reset:
M    hello.html
dknzita@dknzita-VirtualBox:~/hello$ git checkout hello.html
Updated 1 path from the index
dknzita@dknzita-VirtualBox:~/hello$ git status
On branch master
nothing to commit, working tree clean
dknzita@dknzita-VirtualBox:~/hello$

```

Figure 4.19: Работа непосредственно с объектами git

Исследуем git репозиторий вручную самостоятельно. Используя хэш родительского коммита последовательно дойдем до первой версии файла hello.html и посмотрим его(рис. fig. 4.20).

```

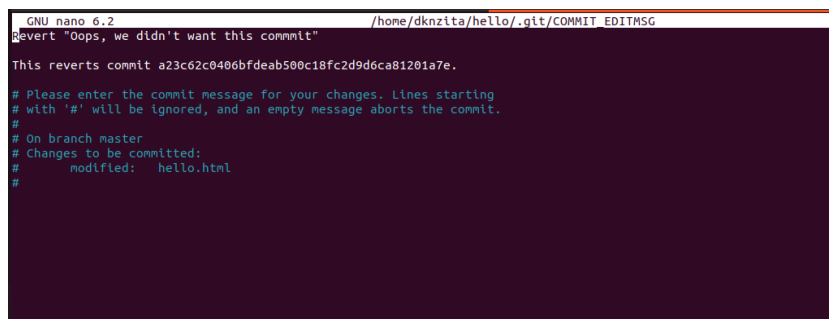
/home/dknzita/hello/hello.html  [----] 11 L: [ 1+ 6 7/ 8] *(129 / 137b) 0010 0x00A
<html>
  <head>
  </head>
  <body>
    <h1> Hello, World! </h1>
    <div> This is an unwanted but committed change -->
  </body>
</html>

```

Figure 4.20: Поиск оригинального файла hello.html

4.17 Создание ветки

Создадим новую ветку «style» и перейдем в неё. Добавим туда файл стилей style.css и добавим его в репозиторий. Обновим файл hello.html, чтобы использовать стили style.css и index.html, также обавим их в репозиторий(рис. fig. 4.21).



```
GNU nano 6.2 /home/dknzita/hello/.git/COMMIT_EDITMSG
Revert "Oops, we didn't want this commit"

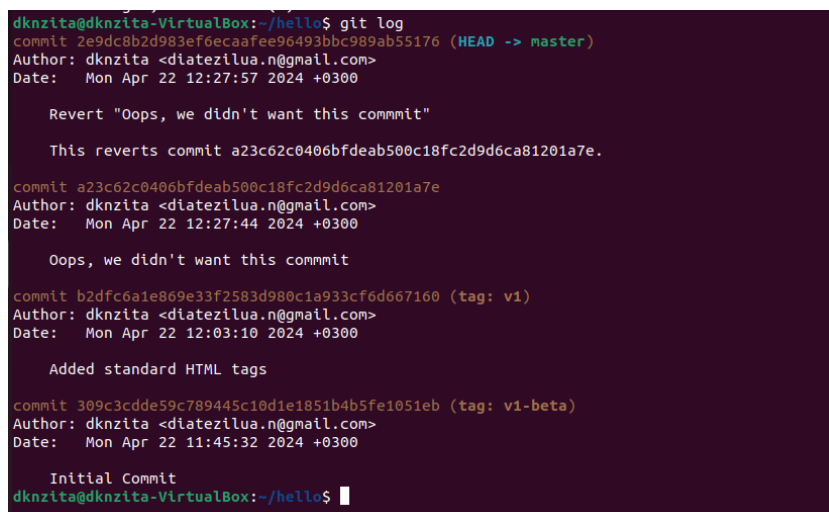
This reverts commit a23c62c0406bfdeab500c18fc2d9d6ca81201a7e.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
#
# Changes to be committed:
#   modified:   hello.html
#
```

Figure 4.21: Создание ветки

4.18 Навигация по веткам

Посмотрим все логи(рис. fig. 4.22).



```
dknzita@dknzita-VirtualBox: ~/hello$ git log
commit 2e9dc8b2d983ef6ecaaf96493bbc989ab55176 (HEAD -> master)
Author: dknzita <diatezilua.n@gmail.com>
Date:   Mon Apr 22 12:27:57 2024 +0300

    Revert "Oops, we didn't want this commit"

    This reverts commit a23c62c0406bfdeab500c18fc2d9d6ca81201a7e.

commit a23c62c0406bfdeab500c18fc2d9d6ca81201a7e
Author: dknzita <diatezilua.n@gmail.com>
Date:   Mon Apr 22 12:27:44 2024 +0300

    Oops, we didn't want this commit

commit b2dfc6a1e869e33f2583d980c1a933cf6d667160 (tag: v1)
Author: dknzita <diatezilua.n@gmail.com>
Date:   Mon Apr 22 12:03:10 2024 +0300

    Added standard HTML tags

commit 309c3cdde59c789445c10d1e1851b4b5fe1051eb (tag: v1-beta)
Author: dknzita <diatezilua.n@gmail.com>
Date:   Mon Apr 22 11:45:32 2024 +0300

    Initial Commit
dknzita@dknzita-VirtualBox: ~/hello$
```

Figure 4.22: Просмотр логов новой ветки

Переключимся обратно на основную ветку и посмотрим содержимое файла

ib/hello.html, заметим, что он не использует стили, также посмотрим содержимое этого файла в новой ветке(рис. fig. 4.23).

```
dknzita@dknzita-VirtualBox:~/hello$ git tag oops
dknzita@dknzita-VirtualBox:~/hello$ git reset --hard v1
HEAD is now at b2dfc6a Added standard HTML tags
dknzita@dknzita-VirtualBox:~/hello$ git log
commit b2dfc6a1e869e33f2583d980c1a933cf6d667160 (HEAD -> master, tag: v1)
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 12:03:10 2024 +0300

    Added standard HTML tags

commit 309c3cdd59c789445c10d1e1851b4b5fe1051eb (tag: v1-beta)
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 11:45:32 2024 +0300

    Initial Commit
dknzita@dknzita-VirtualBox:~/hello$
```

Figure 4.23: Переключение между ветками

4.19 Изменения в ветке master

Вернемся в основную ветку и добавим файл README.md. Посмотрим ветки и их различия(рис. fig. 4.24, fig. 4.25).

```
Initial Commit
dknzita@dknzita-VirtualBox:~/hello$ git log -all
error: switch 'l' expects a numerical value
dknzita@dknzita-VirtualBox:~/hello$ git tag -d oops
Deleted tag 'oops' (was 2e9dc8b)
dknzita@dknzita-VirtualBox:~/hello$ git log --all
commit b2dfc6a1e869e33f2583d980c1a933cf6d667160 (HEAD -> master, tag: v1)
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 12:03:10 2024 +0300

    Added standard HTML tags

commit 309c3cdd59c789445c10d1e1851b4b5fe1051eb (tag: v1-beta)
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 11:45:32 2024 +0300

    Initial Commit
dknzita@dknzita-VirtualBox:~/hello$
```

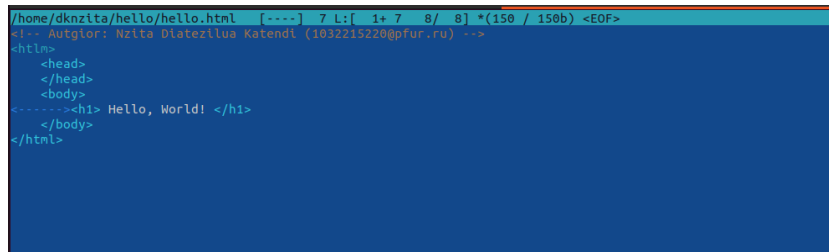
Figure 4.24: Изменения в ветке master

```
/home/dknzita/hello/hello.html [----] 7 L: [ 1+ 7 8/ 8] *(129 / 129b) <EOF>
<!-- Autgior: Nzita Diatezilua Katendi -->
<html>
  <head>
  </head>
  <body>
<-----><h1> Hello, World! </h1>
  </body>
</html>
```

Figure 4.25: Просмотр веток

4.20 Слияние

Слияние переносит изменения из двух веток в одну. Вернемся к ветке style и сольем master с style(рис. fig. 4.26).

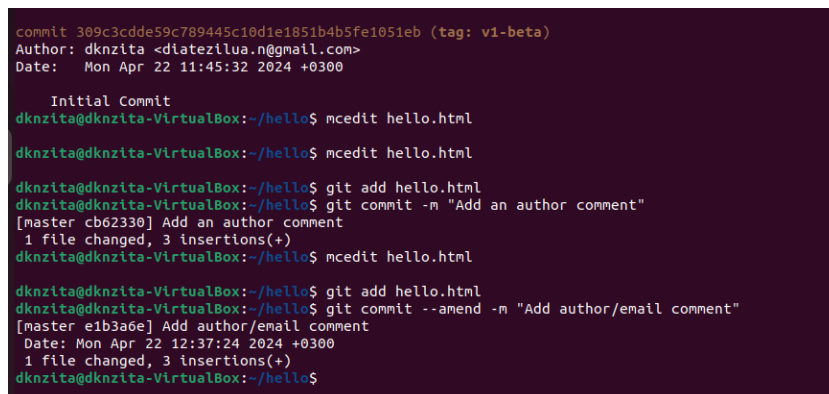


```
/home/dknzita/hello/hello.html {-----} 7 Ls[ 1+ 7 8/ 8] *(150 / 150b) <EOF>
el-- Autgior: Nzita Diatezilua Katendi (1032215220@pfur.ru) -->
<html>
  <head>
  </head>
  <body>
<-----><h1> Hello, World! </h1>
  </body>
</html>
```

Figure 4.26: Слияние веток

4.21 Создание конфликта

Вернемся в ветку master и создадим конфликт, внося изменения в файл hello.html. Просмотрим ветки. После коммита «Added README» ветка master была объединена с веткой style, но в настоящее время в master есть дополнительный коммит, который не был слит с style. Последнее изменение в master конфликтует с некоторыми изменениями в style(рис. fig. 4.27).



```
commit 309c3cdd59c789445c10d1e1851b4b5fe1051eb (tag: v1-beta)
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 11:45:32 2024 +0300

Initial Commit
dknzita@dknzita-VirtualBox:~/hello$ mcedit hello.html
dknzita@dknzita-VirtualBox:~/hello$ mcedit hello.html

dknzita@dknzita-VirtualBox:~/hello$ git add hello.html
dknzita@dknzita-VirtualBox:~/hello$ git commit -m "Add an author comment"
[master cb62330] Add an author comment
1 file changed, 3 insertions(+)
dknzita@dknzita-VirtualBox:~/hello$ mcedit hello.html

dknzita@dknzita-VirtualBox:~/hello$ git add hello.html
dknzita@dknzita-VirtualBox:~/hello$ git commit --amend -m "Add author/email comment"
[master e1b3a6e] Add author/email comment
Date: Mon Apr 22 12:37:24 2024 +0300
1 file changed, 3 insertions(+)
dknzita@dknzita-VirtualBox:~/hello$
```

Figure 4.27: Создание конфликта

4.22 Разрешение конфликтов

Вернемся к ветке `style` и попытаемся объединить ее с новой веткой `master`. В файле `lib/hello.html` можно увидеть записи с обеих версий этого файла. Первый раздел — версия текущей ветки (`style`). Второй раздел — версия ветки `master`. Внесем изменения в `lib/hello.html`, оставив только необходимую нам запись и добавим этот файл в репозиторий, чтобы вручную разрешить конфликт(рис. fig. 4.28).

```
dknzita@dknzita-VirtualBox:~/hello$ mkdir lib
dknzita@dknzita-VirtualBox:~/hello$ git mv hello.html lib
dknzita@dknzita-VirtualBox:~/hello$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    hello.html -> lib/hello.html
dknzita@dknzita-VirtualBox:~/hello$
```

Figure 4.28: Разрешение конфликта

4.23 Сброс ветки `style`

Вернемся на ветке `style` к точке перед тем, как мы слили ее с веткой `master`. Мы хотим вернуться в ветке `style` в точку перед слиянием с `master`. Нам необходимо найти последний коммит перед слиянием(рис. fig. 4.29).

```
dknzita@dknzita-VirtualBox:~/hello$ git commit -m "Moved hello to lib"
[master f4d67b8] Moved hello to lib
1 file changed, 0 insertions(+), 0 deletions(-)
rename hello.html => lib/hello.html (100%)
dknzita@dknzita-VirtualBox:~/hello$
```

Figure 4.29: Поиск коммита перед слиянием

Мы видим, что коммит «Updated index.html» был последним на ветке `style` перед слиянием. Сбросим ветку `style` к этому коммиту(рис. fig. 4.30).


```
/home/dknzita/hello/index.html [-M--] 6 L: 1+ 4 5/ 5] *(96 / 97b) 0062 0x03E
<html>
  <body>
<-----><iframe src = "lib/hello.html" width="200" height="200" />
  </body>
</html>
```

Figure 4.30: Сброс ветки style

Поиским лог ветки style. Увидим, что у нас в истории больше нет коммитов слияний.

4.24 Сброс ветки master

Добавив интерактивный режим в ветку master, мы внесли изменения, конфликтующие с изменениями в ветке style. Давайте вернемся в ветке master в точку перед внесением конфликтующих изменений. Это позволяет нам продемонстрировать работу команды git rebase, не беспокоясь о конфликтах. Просмотрим коммиты ветки master(рис. fig. 4.31).

```
dknzitagdknzita-VirtualBox:~/hello$ git commit -m "Moved hello to lib"
[master f4d67b8] Moved hello to lib
1 file changed, 0 insertions(+), 0 deletions(-)
rename hello.html => lib/hello.html (100%)
dknzitagdknzita-VirtualBox:~/hello$ ls
lib
dknzitagdknzita-VirtualBox:~/hello$ touch index.html
dknzitagdknzita-VirtualBox:~/hello$ mcedit index.html

dknzitagdknzita-VirtualBox:~/hello$ git add index.html
dknzitagdknzita-VirtualBox:~/hello$ git commit -m "Added index.html."
[master a6c0a58] Added index.html.
1 file changed, 5 insertions(+)
create mode 100644 index.html
dknzitagdknzita-VirtualBox:~/hello$ ls -C .git
branches  COMMIT_EDITMSG  config  description  HEAD  hooks  index  info  logs  objects  ORIG_HEAD  packed-refs  refs
dknzitagdknzita-VirtualBox:~/hello$ ls -C .git/objects
ls: cannot access '.git/objects': No such file or directory
dknzitagdknzita-VirtualBox:~/hello$ ls -C .git/objects
10 2e 30 3f 0e 00 97 98 a1 02 a0 ad b2 b4 b8 bd c1 cb cc cd e1 f4 fa info pack
dknzitagdknzita-VirtualBox:~/hello$
```

Figure 4.31: Поиск коммита перед конфликтом

Коммит «Added README» идет непосредственно перед коммитом конфликтующего интерактивного режима. Мы сбросим ветку master к коммиту «Added README»(рис. fig. 4.32).

```
dknznita@dknznita-VirtualBox:~/hello$ ls -C .git
branches  COMMIT_EDITMSG  config  description  HEAD  hooks  index  info  logs  objects  ORIG_HEAD  packed-refs  refs
dknznita@dknznita-VirtualBox:~/hello$ ls -C .git/objects
ls: cannot access '.git/objects': No such file or directory
dknznita@dknznita-VirtualBox:~/hello$ ls -C .git/objects
38 3e 36 37 3e 38 37 39 a1 a2 a6 ad b2 b4 b8 bd c1 cb cc cd e1 f4 fa info pack
dknznita@dknznita-VirtualBox:~/hello$ ls -C .git/objects/cdir>
bash: syntax error near unexpected token 'newline'
dknznita@dknznita-VirtualBox:~/hello$ cat .git/config
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
dknznita@dknznita-VirtualBox:~/hello$ ls .git/refs
heads  tags
dknznita@dknznita-VirtualBox:~/hello$ ls .git/refs/heads
master
dknznita@dknznita-VirtualBox:~/hello$ ls .git/refs/tags
v1  v1-beta
dknznita@dknznita-VirtualBox:~/hello$ ls .git/refs/tags/v1
.git/refs/tags/v1
dknznita@dknznita-VirtualBox:~/hello$ cat .git/HEAD
ref: refs/heads/master
dknznita@dknznita-VirtualBox:~/hello$ git log --max-count=1
commit a6c6a58e91c398ec2929e31cdfa01c3a49382abe (HEAD -> master)
Author: dknznita <diatezilua.n@gmail.com>
Date:   Mon Apr 22 12:46:28 2024 +0300

    Added index.html.
dknznita@dknznita-VirtualBox:~/hello$
```

Figure 4.32: Сброс ветки master

4.25 Перебазирование

Используем команду rebase вместо команды merge. Мы вернулись в точку до первого слияния и хотим перенести изменения из ветки master в нашу ветку style. На этот раз для переноса изменений из ветки master мы будем использовать команду git rebase вместо слияния(рис. fig. 4.33).

```
dknznita@dknznita-VirtualBox:~/hello$ ls -C .git/objects/cdir>
bash: syntax error near unexpected token 'newline'
dknznita@dknznita-VirtualBox:~/hello$ cat .git/config
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
dknznita@dknznita-VirtualBox:~/hello$ ls .git/refs
heads  tags
dknznita@dknznita-VirtualBox:~/hello$ ls .git/refs/heads
master
dknznita@dknznita-VirtualBox:~/hello$ ls .git/refs/tags
v1  v1-beta
dknznita@dknznita-VirtualBox:~/hello$ ls .git/refs/tags/v1
.git/refs/tags/v1
dknznita@dknznita-VirtualBox:~/hello$ cat .git/HEAD
ref: refs/heads/master
dknznita@dknznita-VirtualBox:~/hello$ git log --max-count=1
commit a6c6a58e91c398ec2929e31cdfa01c3a49382abe (HEAD -> master)
Author: dknznita <diatezilua.n@gmail.com>
Date:   Mon Apr 22 12:46:28 2024 +0300

    Added index.html.
dknznita@dknznita-VirtualBox:~/hello$
```

Figure 4.33: Перебазирование

4.26 Слияние в ветку master

Вернемся в ветку master и сольем ветку style в неё с помощью команды git merge(рис. fig. 4.34).

```
/home/dknzita/hello/lib/style.css [-M--] 1 L:[ 1+ 2 3/ 3] *(21 / 21b) <EOF
h1 {
  color: red
}
```

Figure 4.34: Слияние style в master

4.27 Клонирование репозитория

Перейдем в наш рабочий каталог и сделаем клон репозитория hello, затем создадим клон репозитория. Просмотрев его увидим список всех файлов на верхнем уровне оригинального репозитория README.md, index.html и lib. Затем посмотрим историю репозитория и увидим список всех коммитов в новый репозиторий, и он совпадает с историей коммитов в оригинальном репозитории. Единствен в названиях веток(рис. fig. 4.35).

```
Added index.html.
dknzita@dknzita-VirtualBox:~/hello$ git checkout -b style
Switched to a new branch 'style'
dknzita@dknzita-VirtualBox:~/hello$ git status
On branch style
nothing to commit, working tree clean
dknzita@dknzita-VirtualBox:~/hello$ touch lib/style.css
dknzita@dknzita-VirtualBox:~/hello$ mcedit lib/style.css

dknzita@dknzita-VirtualBox:~/hello$ git add lib/style.css
dknzita@dknzita-VirtualBox:~/hello$ git commit -m "Added css stylesheet"
[style c2df971] Added css stylesheet
1 file changed, 3 insertions(+)
create mode 100644 lib/style.css
dknzita@dknzita-VirtualBox:~/hello$
```

Figure 4.35: Клонирование репозитория

4.28 Что такое origin?

Клонированный репозиторий знает об имени по умолчанию удаленного репозитория. Посмотрим, подробную информацию об имени по умолчанию. Для того, чтобы увидеть все ветки используем опцию -a(рис. fig. 4.36).

```
/home/dknzita/hello/hello.html [-----] 6 L: 1 1+10 11/ 11] *(234 / 234b) <EOF>
<!-- Author: Nizta Blatezilua Katendi Nizta (1032215220qqpfur.ru) -->

<html>
  <head>
    <-----><link type = "text/css" rel="stylesheet"
    <-----><meta= all href= "style.css" />
    </head>
    <body>
      <h1> Hello, world! </h1>
    </body>
  </html>
```

Figure 4.36: Просмотр имени по умолчанию удаленного репозитория

4.29 Удаленные ветки

Посмотрим на ветки, доступные в нашем клонированном репозитории. Можно увидеть, что в списке только ветка master(рис. fig. 4.37).

```
dknzita@dknzita-VirtualBox:~/hello$ touch lib/style.css
dknzita@dknzita-VirtualBox:~/hello$ mcedit lib/style.css

dknzita@dknzita-VirtualBox:~/hello$ git add lib/style.css
dknzita@dknzita-VirtualBox:~/hello$ git commit -m "Added css stylesheet"
[style c2df971] Added css stylesheet
1 file changed, 3 insertions(+)
create mode 100644 lib/style.css
dknzita@dknzita-VirtualBox:~/hello$ touch hello.html
dknzita@dknzita-VirtualBox:~/hello$ mcedit hello.html

dknzita@dknzita-VirtualBox:~/hello$ git add lib/hello.html
dknzita@dknzita-VirtualBox:~/hello$ git commit -m "Hello uses style.css"
On branch style
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    hello.html

nothing added to commit but untracked files present (use "git add" to track)
dknzita@dknzita-VirtualBox:~/hello$
```

Figure 4.37: Просмтр доступных веток

4.30 Изменение оригинального репозитория

Перейдем в репозиторий hello. Внесем изменения в файл README.md. Затем добавим их в репозиторий(рис. fig. 4.38).

```
dknzita@dknzita-VirtualBox:~/hello$ git log --all
commit c2df971dda80a73fb5a8aac7389e95a20addf428 (HEAD -> style)
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 12:56:50 2024 +0300

    Added css stylesheet

commit a6c6a58e91c398ec2929e31cdfa01c3a49382abe (master)
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 12:46:28 2024 +0300

    Added index.html.

commit f4d67b898f83f15989f9b79d90941cdba2a82538
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 12:41:48 2024 +0300

    Moved hello to lib

commit e1b3a6e612fec1e0c2081e79c1201ec558f12727
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 12:37:24 2024 +0300

    Add author/email comment

commit b2dfc6a1e869e33f2583d980c1a933cf6d667160 (tag: v1)
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 12:03:10 2024 +0300

    Added standard HTML tags

commit 309c3cdd59c709445c10d1e1851b4b5fe1051eb (tag: v1-beta)
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 11:45:32 2024 +0300

    Initial Commit
dknzita@dknzita-VirtualBox:~/hello$
```

Figure 4.38: Изменение оригинального репозитория

Перейдём в клон репозитория и используем команду `git fetch`, которая будет извлекать новые коммиты из удаленного репозитория, но не будет сливать их с наработками в локальных ветках(рис. fig. 4.39).

```
Initial Commit
dknzita@dknzita-VirtualBox:~/hello$ git checkout master
Switched to branch 'master'
dknzita@dknzita-VirtualBox:~/hello$ cat lib/hello.html
<!-- Autgior: Nzita Diatezilua Katendi (1032215220@pfur.ru) -->
<html>
<head>
</head>
<body>
    <h1> Hello, World! </h1>
</body>
</html>
dknzita@dknzita-VirtualBox:~/hello$
```

Figure 4.39: Извлечение изменений

4.31 Слияние извлеченных изменений

Сольем внесённые изменения в главную ветку. Также можно было бы использовать команду `git pull`, которая является объединением `fetch` и `merge` в одну команду.

(рис. fig. 4.40).

```

Initial Commit
dknzita@dknzita-VirtualBox:~/hello$ git checkout master
Switched to branch 'master'
dknzita@dknzita-VirtualBox:~/hello$ cat lib/hello.html
<!-- Autgior: Nzita Diatezilua Katendi (1032215220@pfur.ru) -->
<html>
  <head>
  </head>
  <body>
    <h1> Hello, World! </h1>
  </body>
</html>
dknzita@dknzita-VirtualBox:~/hello$ git checkout master
Already on 'master'
dknzita@dknzita-VirtualBox:~/hello$ touch README.md
dknzita@dknzita-VirtualBox:~/hello$ echo "This is the hello Word example from git tutorial." > README.md
dknzita@dknzita-VirtualBox:~/hello$ git add README.md
dknzita@dknzita-VirtualBox:~/hello$ git commit -m "Added README"
[master d59ddff] Added README
1 file changed, 1 insertion(+)
create mode 100644 README.md
dknzita@dknzita-VirtualBox:~/hello$

```

Figure 4.40: Слияние извлеченных изменений

4.32 Добавление ветки наблюдения

Добавим локальную ветку, которая отслеживает удаленную ветку, теперь мы можем видеть ветку style в списке веток и логе(рис. fig. 4.41).

```

dknzita@dknzita-VirtualBox:~/hello$ git log --graph --all
* commit d59ddffab8d4ab2ab9270b6ee456aa3a65c461 (HEAD -> master)
  Author: dknzita <diatezilua.n@gmail.com>
  Date: Mon Apr 22 13:09:30 2024 +0300

    Added README

* commit d59ddffab8d4ab2ab9270b6ee456aa3a65c461 (HEAD -> master)
  Author: dknzita <diatezilua.n@gmail.com>
  Date: Mon Apr 22 13:09:30 2024 +0300

    Added README

* commit c2df971dda80a73fb5a8aac7389e95a20addf428 (style)
  Author: dknzita <diatezilua.n@gmail.com>
  Date: Mon Apr 22 12:56:50 2024 +0300

    Added css stylesheet

* commit a6c6a58e91c398ec2929e31cdfa01c3a49382abe
  Author: dknzita <diatezilua.n@gmail.com>
  Date: Mon Apr 22 12:46:28 2024 +0300

    Added index.html.

* commit f4d67b898f83f15989f9b79d90941cd8a2a82538
  Author: dknzita <diatezilua.n@gmail.com>
  Date: Mon Apr 22 12:41:48 2024 +0300

```

Figure 4.41: Добавление ветки наблюдения

4.33 Создание чистого репозитория

Как правило, репозитории, оканчивающиеся на .git являются чистыми репозиториями. Создадим такой в рабочем каталоге. Затем добавим репозиторий

hello.git к нашему оригинальному репозиторию(рис. fig. 4.42).

```
* commit d59ddffbad0d4ab2ab9270b0ee456aa3a65c461 (HEAD -> master)
Author: dknzita <diatezilua.ng@gmail.com>
Date: Mon Apr 22 13:09:30 2024 +0300

    Added README

* commit c2df971dda90a73fb5a8aac7389e95a20addf428 (style)
/ Author: dknzita <diatezilua.ng@gmail.com>
Date: Mon Apr 22 12:56:50 2024 +0300

    Added css stylesheet

* commit a6c6a58e91c398ec2979e31cdfa01c3a49382abe
Author: dknzita <diatezilua.ng@gmail.com>
Date: Mon Apr 22 12:46:28 2024 +0300

    Added index.html.

* commit f4d67b898f83f15989f9b79d90941cda2a82538
Author: dknzita <diatezilua.ng@gmail.com>
Date: Mon Apr 22 12:41:48 2024 +0300

    Moved hello to lib

* commit e1b3a6e012fec1e0c2081e79c1201ec558f12727
Author: dknzita <diatezilua.ng@gmail.com>
Date: Mon Apr 22 12:37:24 2024 +0300

    Add author/email comment

* commit b2dfc0a1e88e33f2383d880c1a933cf0d067160 (tag: v1)
Author: dknzita <diatezilua.ng@gmail.com>
Date: Mon Apr 22 12:03:10 2024 +0300

    Added standard HTML tags
```

Figure 4.42: Создание чистого репозитория

4.34 Отправка и извлечение изменений

Так как чистые репозитории, как правило, расшариваются на каком-нибудь сетевом сервере, нам необходимо отправить наши изменения в другие репозитории. Начнем с создания изменения для отправки. Отредактируем файл README.md и сделаем коммит, затем отправим изменения в общий репозиторий. Затем извлечем изменения из общего репозитория(рис. fig. 4.43).

```

* commit 9d1dd3b17b70fa4c7bb2d160323ad6edd20b437b (HEAD -> style)
Merge: c2df971 d59ddff
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 13:13:15 2024 +0300

    Merge branch 'master' into style

* commit d59ddffab8d4ab2ab9270b6ee456aa3a65c461 (master)
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 13:09:30 2024 +0300

    Added README

* commit c2df971dda80a73fb5a8aac7389e95a20addf428
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 12:56:50 2024 +0300

    Added css stylesheet

* commit a6c6a58e91c398ec2929e31cdfa01c3a49382abe
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 12:46:28 2024 +0300

    Added index.html.

* commit f4d67b898f83f15989f9b79d98941cd8a2a82538
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 12:41:48 2024 +0300

    Moved hello to lib

* commit e1b3a6e612fec1e0c2081e79c1201ec558f12727
Author: dknzita <diatezilua.n@gmail.com>
Date: Mon Apr 22 12:37:24 2024 +0300

    Add author/email comment
:

```

Figure 4.43: Отправка и извлечение изменений

5 Выводы

В результате выполнения лабораторной работы были приобретены практические навыки работы с системой управления версиями Git.