

Лабораторная работа №1

Работа с git

Акондзо Жордани Лади Гаэл

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Подготовка	6
3.2	Создание проекта	6
3.3	Внесение изменений	7
3.4	История	9
3.5	Получение старых версий	10
3.6	Создание тегов версий	11
3.7	Отмена локальных изменений (до индексации)	13
3.8	Отмена проиндексированных изменений (перед коммитом)	13
3.9	Отмена коммитов	14
3.10	Удаление коммитов из ветки	16
3.11	Удаление тега оорс	16
3.12	Изменение предыдущего коммита	17
3.13	Перемещение файлов	18
3.14	Подробнее о структуре	19
3.15	Git внутри: Каталог .git	20
3.16	Работа непосредственно с объектами git	20
3.17	Создание ветки	22
3.18	Навигация по веткам	22
4	Выводы	24

Список иллюстраций

3.1	Настройка git	6
3.2	Создание репозитория	7
3.3	Внесение изменений в содержимое репозитория	8
3.4	Внесение нескольких изменений в содержимое репозитория	9
3.5	Просмотр истории	10
3.6	Просмотр разных версий репозитория	11
3.7	Создание тегов версий	12
3.8	Переключение по имени тега и просмотр доступных тегов	12
3.9	Отмена локальных изменений (до индексации)	13
3.10	Отмена проиндексированных изменений (перед коммитом)	14
3.11	Отмена коммитов	15
3.12	Удаление коммитов из ветки	16
3.13	Удаление тега оорс	17
3.14	Изменение предыдущего коммита	18
3.15	Перемещение файлов	18
3.16	Добавление нового файла в репозиторий	19
3.17	Добавление нового файла в репозиторий	19
3.18	Каталог .git	20
3.19	Работа непосредственно с объектами git	21
3.20	Поиск оригинального файла hello.html	21
3.21	Создание ветки	22
3.22	Просмотр логов новой ветки	23

1 Цель работы

Приобретение практических навыков работы с ситемой управления версиями Git.

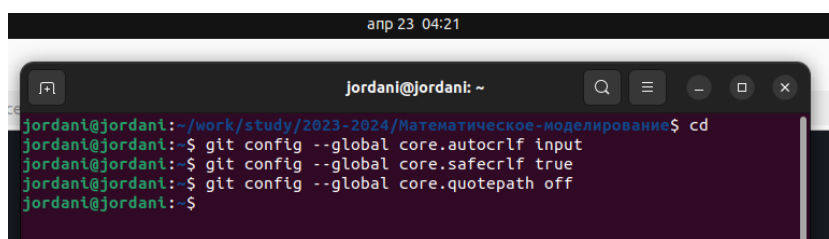
2 Задание

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

3 Выполнение лабораторной работы

3.1 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. 3.1).

A screenshot of a terminal window with a dark background. The window title is "Jordani@Jordani: ~". The terminal shows the following commands and their outputs:

```
Jordani@Jordani:~/work/study/2023-2024/Математическое-моделирование$ cd
Jordani@Jordani:~$ git config --global core.autocrlf input
Jordani@Jordani:~$ git config --global core.safecrlf true
Jordani@Jordani:~$ git config --global core.quotePath off
Jordani@Jordani:~$
```

Рис. 3.1: Настройка git

3.2 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. 3.2).

```
les Terminal anp 23 04:33 Jordani@Jordani: ~/hello

Jordani@Jordani:~$ mkdir hello
Jordani@Jordani:~$ cd hello
Jordani@Jordani:~/hello$ touch hello.html
Jordani@Jordani:~/hello$ echo "Hello, World!" > hello.html
Jordani@Jordani:~/hello$ cat hello
cat: hello: No such file or directory
Jordani@Jordani:~/hello$ cat hello.html
Hello, World!
Jordani@Jordani:~/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:
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:
hint:   git branch -m <name>
Initialized empty Git repository in /home/Jordani/hello/.git/
Jordani@Jordani:~/hello$ git add hello.html
Jordani@Jordani:~/hello$ git commit -m "Initial Commit"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'Jordani@Jordani.(none)')
Jordani@Jordani:~/hello$ git config --global user.name "Jordaniakondzo"
Jordani@Jordani:~/hello$ git config --global user.email "jordaniakondzo3438@gmail.com"
Jordani@Jordani:~/hello$ git add hello.html
Jordani@Jordani:~/hello$ git commit -m "Initial Commit"
[master (root-commit) 39958c5] Initial Commit
1 file changed, 1 insertion(+)
create mode 100644 hello.html
Jordani@Jordani:~/hello$ git status
On branch master
nothing to commit, working tree clean
Jordani@Jordani:~/hello$
```

Рис. 3.2: Создание репозитория

3.3 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. 3.3).

```
les Terminal anp 23 04:42
Jordani@Jordani: ~/he

Jordani@Jordani:~/hello$ nano hello.html
Jordani@Jordani:~/hello$ cat hello.html
<h1>Hello, World!</h1>
Jordani@Jordani:~/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")
Jordani@Jordani:~/hello$ git add hello.html
Jordani@Jordani:~/hello$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   hello.html

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

Jordani@Jordani:~/hello$ git commit -m "adding"
[master ff4beef] adding
1 file changed, 1 insertion(+), 1 deletion(-)
Jordani@Jordani:~/hello$ git status
On branch master
nothing to commit, working tree clean
Jordani@Jordani:~/hello$
```

Рис. 3.3: 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. 3.4).


```
les Terminal anp 23 04:49
Jordanl@Jordanl: ~/h

Jordanl@Jordanl:~/hello$ nano hello.html
Jordanl@Jordanl:~/hello$ cat hello.html
<html>
<body>
<h1>Hello, World!</h1>
</body>
</html>

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

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

Jordanl@Jordanl:~/hello$ git commit -m "Added standard HTML page tags"
[master ebfd4b] Added standard HTML page tags
1 file changed, 5 insertions(+)
Jordanl@Jordanl:~/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")
Jordanl@Jordanl:~/hello$ git add .
Jordanl@Jordanl:~/hello$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

Jordanl@Jordanl:~/hello$ git commit -m "Added HTML header"
[master b2fa60b] Added HTML header
1 file changed, 2 insertions(+), 1 deletion(-)
Jordanl@Jordanl:~/hello$
```

Рис. 3.4: Внесение нескольких изменений в содержимое репозитория

3.4 История

Посмотрим список произведённых изменений в стандартном виде, затем в однострочном, а также с указанием времени и количества (рис. fig. 3.5).

```
ties Terminal anp 23 04:55
Jordanl@Jordan: ~/hello

jordanl@jordan:~/hello$ git log
commit b2fa60b0336f33e373591c0d9314c571de0f413 (HEAD -> master)
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:49:10 2024 +0300

    Added HTML header

commit ebfd04b1ad859356256ccc14248c29842fbaee54
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:47:24 2024 +0300

    Added standard HTML page tags

commit ff4beef7bd3f7647fe69eca37f5a947616ef15c4
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:41:49 2024 +0300

    adding

commit 39958c5c335173e87fceb1b5d4159139d4aae245
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:31:47 2024 +0300

    Initial Commit

jordanl@jordan:~/hello$ git log --pretty=oneline
b2fa60b0336f33e373591c0d9314c571de0f413 (HEAD -> master) Added HTML header
ebfd04b1ad859356256ccc14248c29842fbaee54 Added standard HTML page tags
ff4beef7bd3f7647fe69eca37f5a947616ef15c4 adding
39958c5c335173e87fceb1b5d4159139d4aae245 Initial Commit
jordanl@jordan:~/hello$ git log --pretty=oneline --max-count=2
b2fa60b0336f33e373591c0d9314c571de0f413 (HEAD -> master) Added HTML header
ebfd04b1ad859356256ccc14248c29842fbaee54 Added standard HTML page tags
jordanl@jordan:~/hello$ git log --pretty=oneline --since='5 minutes ago'
b2fa60b0336f33e373591c0d9314c571de0f413 (HEAD -> master) Added HTML header
ebfd04b1ad859356256ccc14248c29842fbaee54 Added standard HTML page tags
jordanl@jordan:~/hello$ git log --pretty=oneline --until='5 minutes ago'
ff4beef7bd3f7647fe69eca37f5a947616ef15c4 adding
39958c5c335173e87fceb1b5d4159139d4aae245 Initial Commit
jordanl@jordan:~/hello$ git log --pretty=oneline --author=Jordaniakondzo
b2fa60b0336f33e373591c0d9314c571de0f413 (HEAD -> master) Added HTML header
ebfd04b1ad859356256ccc14248c29842fbaee54 Added standard HTML page tags
ff4beef7bd3f7647fe69eca37f5a947616ef15c4 adding
39958c5c335173e87fceb1b5d4159139d4aae245 Initial Commit
jordanl@jordan:~/hello$ git log --pretty=oneline --all
b2fa60b0336f33e373591c0d9314c571de0f413 (HEAD -> master) Added HTML header
ebfd04b1ad859356256ccc14248c29842fbaee54 Added standard HTML page tags
ff4beef7bd3f7647fe69eca37f5a947616ef15c4 adding
39958c5c335173e87fceb1b5d4159139d4aae245 Initial Commit
```

Рис. 3.5: Просмотр истории

3.5 Получение старых версий

Изучим данные лога и найдем там хэш первого коммита, используя его вернемся к первой версии и просмотрим файл `hello.html`, действительно, увидим первую версию. Затем вернемся к последней версии в ветке `master` и вновь посмотрим на файл(рис. fig. 3.6).

```
ties Terminal anp 23 05:02
JordanI@Jordan: ~/hello

Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:41:49 2024 +0300

adding
commit 39958c5c335173e87fceb1b5d4159139d4aae245
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:31:47 2024 +0300

Initial Commit
JordanI@Jordan:~/hello$ git checkout b2fa60b033
Note: switching to 'b2fa60b033'.

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 b2fa60b Added HTML header
JordanI@Jordan:~/hello$ cat hello.html
<html>
<head>
</head>
<body>
<h1>Hello, World!</h1>
</body>
</html>
JordanI@Jordan:~/hello$ git checkout master
Switched to branch 'master'
JordanI@Jordan:~/hello$ cat hello.html
<html>
<head>
</head>
<body>
<h1>Hello, World!</h1>
</body>
</html>
JordanI@Jordan:~/hello$
```

Рис. 3.6: Просмотр разных версий репозитория

3.6 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. 3.7).

```
es Terminal anp 23 05:05
Jordani@Jordani: ~/hello

Jordani@Jordani:~/hello$ git tag v1
Jordani@Jordani:~/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 ebf0d4b Added standard HTML page tags
Jordani@Jordani:~/hello$ cat hello.html
<html>
<body>
<h1>Hello, World!</h1>
</body>
</html>

Jordani@Jordani:~/hello$ git tag v1-beta
Jordani@Jordani:~/hello$
```

Рис. 3.7: Создание тегов версий

Переключимся по тегам между двумя отмеченными версиями. Просмотрим все доступные теги(их два) и посмотрим теки в логе(рис. fig. 3.8).

```
es Terminal anp 23 05:07
Jordani@Jordani: ~/hello

Jordani@Jordani:~/hello$ git checkout v1
Previous HEAD position was ebf0d4b Added standard HTML page tags
HEAD is now at b2fa60b Added HTML header
Jordani@Jordani:~/hello$ git checkout v1-beta
Previous HEAD position was b2fa60b Added HTML header
HEAD is now at ebf0d4b Added standard HTML page tags
Jordani@Jordani:~/hello$ git tag
v1
v1-beta
Jordani@Jordani:~/hello$ git log master --all
commit b2fa60ba336f32e373591cd9314c571de0f413 (tag: v1, master)
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:49:10 2024 +0300

    Added HTML header

commit ebf0d4b1ad859356256ccc14248c29842fbaee54 (HEAD, tag: v1-beta)
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:47:24 2024 +0300

    Added standard HTML page tags

commit ff4beef7bd3f7647fe69eca37f5a947616ef15c4
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:41:49 2024 +0300

    adding

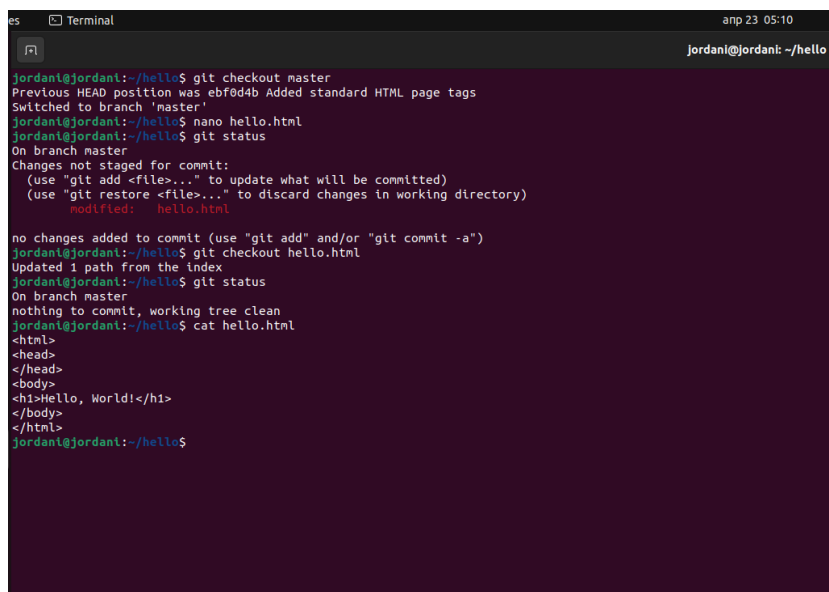
commit 39958c5c335173e87fceb1b5d4159139d4aae245
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:31:47 2024 +0300

    Initial Commit
Jordani@Jordani:~/hello$
```

Рис. 3.8: Переключение по имени тега и просмотр доступных тегов

3.7 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. 3.9).



```
Jordan@Jordan: ~/hello
Jordan@Jordan:~/hello$ git checkout master
Previous HEAD position was ebf0d4b Added standard HTML page tags
Switched to branch 'master'
Jordan@Jordan:~/hello$ nano hello.html
Jordan@Jordan:~/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")
Jordan@Jordan:~/hello$ git checkout hello.html
Updated 1 path from the index
Jordan@Jordan:~/hello$ git status
On branch master
nothing to commit, working tree clean
Jordan@Jordan:~/hello$ cat hello.html
<html>
<head>
</head>
<body>
<h1>Hello, world!</h1>
</body>
</html>
Jordan@Jordan:~/hello$
```

Рис. 3.9: Отмена локальных изменений (до индексации)

3.8 Отмена проиндексированных изменений (перед коммитом)

Внесем изменение в файл hello.html в виде нежелательного комментария

```
<html>

<head>

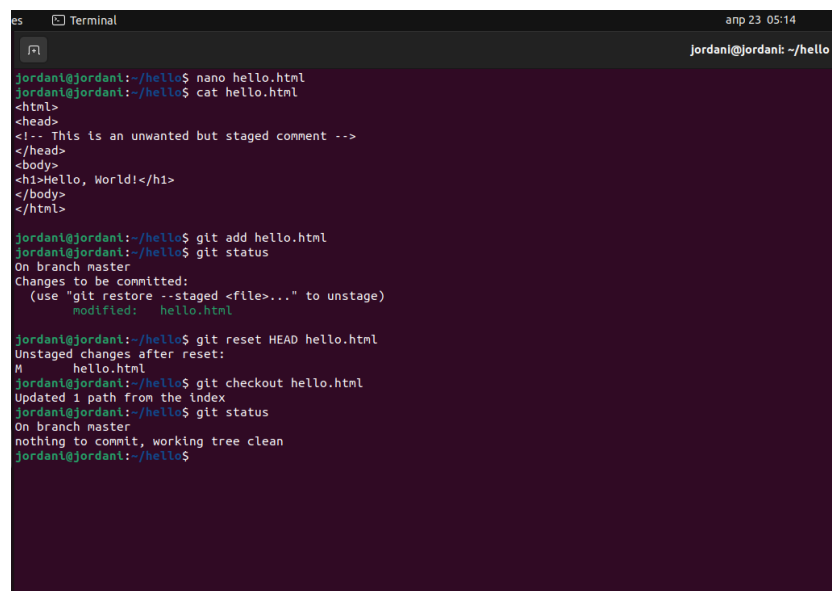
  <!-- This is an unwanted but staged comment -->

</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 команду `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. 3.10).



```
es  Terminal  anp 23 05:14
Jordani@Jordani: ~/hello

Jordani@Jordani:~/hello$ nano hello.html
Jordani@Jordani:~/hello$ cat hello.html
<html>
<head>
<!-- This is an unwanted but staged comment -->
</head>
<body>
<h1>Hello, World!</h1>
</body>
</html>

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

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

Рис. 3.10: Отмена проиндексированных изменений (перед коммитом)

3.9 Отмена коммитов

Изменим файл `hello.html` на следующий.

```
<html>
<head>
</head>
```

```

<body>

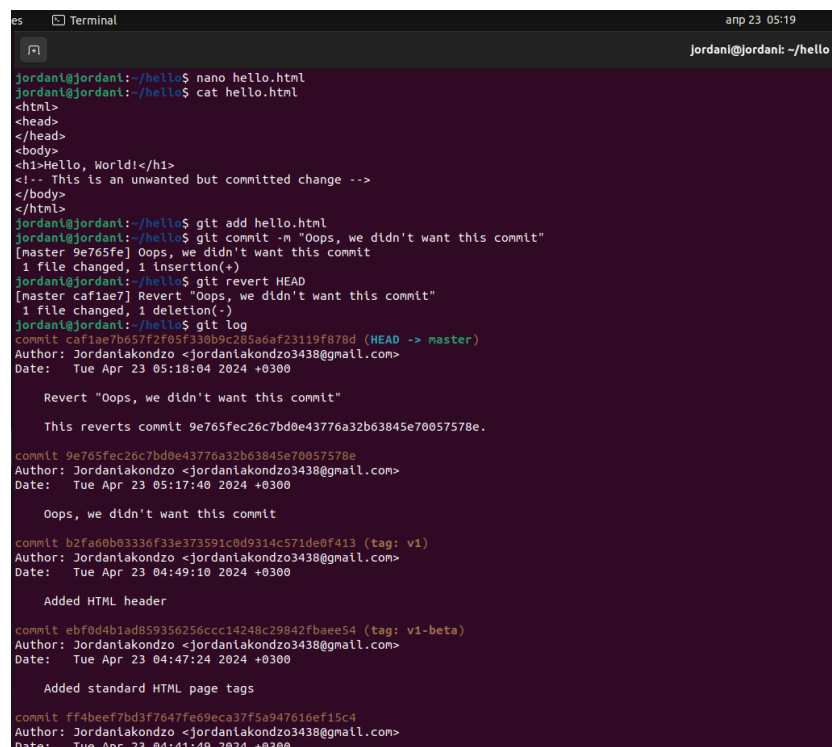
  <h1>Hello, World!</h1>

  <!-- 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, сохраненные нежелательным коммитом. Перейдем в редактор, где изменим нежелательный коммит. Проверим лог. Проверка лога показывает нежелательные и отмененные коммиты в наш репозиторий(рис. fig. 3.11).



```

es  Terminal  anp 23 05:19
JordanI@JordanI: ~/hello

JordanI@JordanI:~/hello$ nano hello.html
JordanI@JordanI:~/hello$ cat hello.html
<html>
<head>
</head>
<body>
<h1>Hello, World!</h1>
<!-- This is an unwanted but committed change -->
</body>
</html>
JordanI@JordanI:~/hello$ git add hello.html
JordanI@JordanI:~/hello$ git commit -m "Oops, we didn't want this commit"
[master 9e765fe] Oops, we didn't want this commit
1 file changed, 1 insertion(+)
JordanI@JordanI:~/hello$ git revert HEAD
[master cafa1ae7] Revert "Oops, we didn't want this commit"
1 file changed, 1 deletion(-)
JordanI@JordanI:~/hello$ git log
commit cafa1ae7b657f2f05f330b9c285a6af23119f878d (HEAD -> master)
Author: JordanIakondzo <jordanIakondzo3438@gmail.com>
Date: Tue Apr 23 05:18:04 2024 +0300

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

    This reverts commit 9e765fec26c7bd0e43776a32b63845e70057578e.

commit 9e765fec26c7bd0e43776a32b63845e70057578e
Author: JordanIakondzo <jordanIakondzo3438@gmail.com>
Date: Tue Apr 23 05:17:40 2024 +0300

    Oops, we didn't want this commit

commit b2fa0b03336f33e373591c0d9314c571de0f413 (tag: v1)
Author: JordanIakondzo <jordanIakondzo3438@gmail.com>
Date: Tue Apr 23 04:49:10 2024 +0300

    Added HTML header

commit ebf0d4b1ad859356256ccc14248c29842fbaee54 (tag: v1-beta)
Author: JordanIakondzo <jordanIakondzo3438@gmail.com>
Date: Tue Apr 23 04:47:24 2024 +0300

    Added standard HTML page tags

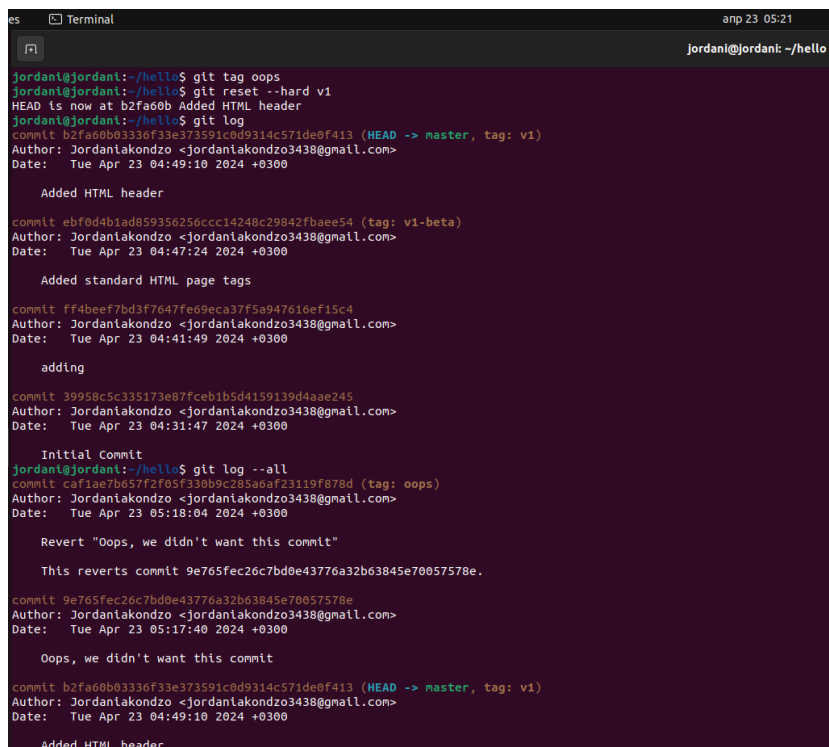
commit ff4beef7bd3f7647fe69eca37f5a947616ef15c4
Author: JordanIakondzo <jordanIakondzo3438@gmail.com>
Date: Tue Apr 23 04:41:49 2024 +0300

```

Рис. 3.11: Отмена коммитов

3.10 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. 3.12).



```
JordanI@JordanI: ~/hello
JordanI@JordanI:~/hello$ git tag oops
JordanI@JordanI:~/hello$ git reset --hard v1
HEAD is now at b2fa60b Added HTML header
JordanI@JordanI:~/hello$ git log
commit b2fa60b03336f33e373591c0d9314c571de0f413 (HEAD -> master, tag: v1)
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:49:10 2024 +0300

    Added HTML header

commit ebf0d4b1ad859356256ccc14248c29842fbaee54 (tag: v1-beta)
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:47:24 2024 +0300

    Added standard HTML page tags

commit ff4beef7bd3f7647fe69eca37f5a947616ef15c4
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:41:49 2024 +0300

    adding

commit 39958c5c335173e07fceb1b5d4159139d4aae245
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:31:47 2024 +0300

    Initial Commit
JordanI@JordanI:~/hello$ git log --all
commit cafae7b657f2f05f330b9c285a6af23119f878d (tag: oops)
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 05:18:04 2024 +0300

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

    This reverts commit 9e765fec26c7bd0e43776a32b63845e70057578e.

commit 9e765fec26c7bd0e43776a32b63845e70057578e
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 05:17:40 2024 +0300

    Oops, we didn't want this commit

commit b2fa60b03336f33e373591c0d9314c571de0f413 (HEAD -> master, tag: v1)
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:49:10 2024 +0300

    Added HTML header
```

Рис. 3.12: Удаление коммитов из ветки

3.11 Удаление тега оорс

Удалим тег оорс и коммиты, на которые он ссылался, сборщиком мусора. Теперь этот тег не отображается в репозитории(рис. fig. 3.13).


```
es Terminal anp 23 05:23
Jordanl@Jordanl: ~/hello
Jordanl@Jordanl:~/hello$ git tag -d oops
Deleted tag 'oops' (was caf1ae7)
Jordanl@Jordanl:~/hello$ git log --all
commit b2fa60b03336f33e373591c0d9314c571de0f413 (HEAD -> master, tag: v1)
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:49:10 2024 +0300

    Added HTML header

commit ebfd4b1ad859356256ccc14240c29842fbaee54 (tag: v1-beta)
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:47:24 2024 +0300

    Added standard HTML page tags

commit ff4beef7bd3f7647fe69eca37f5a947616ef15c4
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:41:49 2024 +0300

    adding

commit 39958c5c335173e07fceb1b5d4159139d4aae245
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:31:47 2024 +0300

    Initial Commit
Jordanl@Jordanl:~/hello$
```

Рис. 3.13: Удаление тега oops

3.12 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. 3.14).

```
JordanI@Jordan: ~/hello
JordanI@Jordan:~/hello$ nano hello.html
JordanI@Jordan:~/hello$ cat hello.html
<!-- Author: Dmitry S. Kulyabov -->
<html>
<head>
</head>
<body>
<h1>Hello, World!</h1>
</body>
</html>

JordanI@Jordan:~/hello$ git add hello.html
JordanI@Jordan:~/hello$ git commit -m "Add an author comment"
[master 952f605] Add an author comment
1 file changed, 2 insertions(+)
JordanI@Jordan:~/hello$ nano hello.html
JordanI@Jordan:~/hello$ git add hello.html
JordanI@Jordan:~/hello$ git commit --amend -m "Add an author/email comment"
[master 8909a5f] Add an author/email comment
Date: Tue Apr 23 05:25:12 2024 +0300
1 file changed, 2 insertions(+)
JordanI@Jordan:~/hello$ git log
commit 8909a5f9427a1b520b32a5367da5a4ec239d0324 (HEAD -> master)
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 05:25:12 2024 +0300

    Add an author/email comment

commit b2fa60b03336f33e373591c0d9314c571de0f413 (tag: v1)
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:49:10 2024 +0300

    Added HTML header

commit ebf0d4b1ad859356256ccc14248c29842fbaee54 (tag: v1-beta)
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:47:24 2024 +0300

    Added standard HTML page tags

commit ff4beef7bd3f7647fe69eca37f5a947616ef15c4
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:41:49 2024 +0300

    adding

commit 3005025c335173e87fceb1b5d4150130d4aap245
```

Рис. 3.14: Изменение предыдущего коммита

3.13 Перемещение файлов

Переместим наш файл в каталог lib. Для этого создадим его и используем команду `git mv`, сделаем коммит этого перемещения(рис. fig. 3.15).

```
es Terminal anp 23 05:32
JordanI@Jordan: ~/hello
JordanI@Jordan:~/hello$ mkdir lib
JordanI@Jordan:~/hello$ git mv hello.html lib
JordanI@Jordan:~/hello$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    hello.html -> lib/hello.html

JordanI@Jordan:~/hello$ git commit -m "Moved hello.html to lib"
[master a4dfec7] Moved hello.html to lib
1 file changed, 0 insertions(+), 0 deletions(-)
rename hello.html => lib/hello.html (100%)
JordanI@Jordan:~/hello$
```

Рис. 3.15: Перемещение файлов

3.14 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. 3.16).

A terminal window titled 'Terminal' with a dark background. The prompt is 'Jordan@jordan: ~/hello'. The user enters the following commands: 'touch index.html', 'nano index.html', 'cat index.html', 'git add index.html', and 'git commit -m "Added index.html."'. The output shows the file being added to the index and committed to the master branch.

```
Jordan@jordan:~/hello$ touch index.html
Jordan@jordan:~/hello$ nano index.html
Jordan@jordan:~/hello$ cat index.html
<html>
<body>
<iframe src="lib/hello.html" width="200" height="200" />
</body>
</html>
Jordan@jordan:~/hello$ git add index.html
Jordan@jordan:~/hello$ git commit -m "Added index.html."
[master d88616a] Added index.html.
1 file changed, 5 insertions(+)
create mode 100644 index.html
Jordan@jordan:~/hello$
```

Рис. 3.16: Добавление нового файла в репозиторий

Теперь при открытии index.html, увидим кусок страницы hello в маленьком окошке(рис. fig. 3.17).

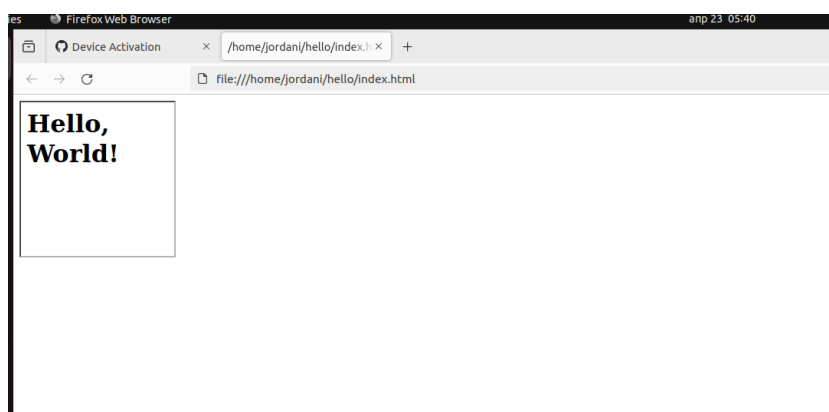
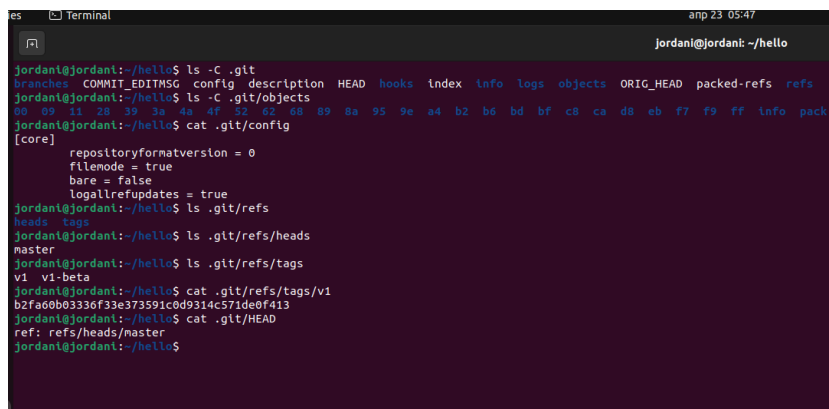


Рис. 3.17: Добавление нового файла в репозиторий

3.15 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. 3.18).



```
Jordan@Jordan: ~/hello
Jordan@Jordan:~/hello$ ls -C .git
branches  COMMIT_EDITMSG  config  description  HEAD  hooks  index  info  logs  objects  ORIG_HEAD  packed-refs  refs
Jordan@Jordan:~/hello$ ls -C .git/objects
00 09 11 28 39 3a 4e 4f 52 52 68 89 8a 95 9e a4 b2 b6 bd bf c8 ca d8 eb f7 f9 ff info pack
Jordan@Jordan:~/hello$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
Jordan@Jordan:~/hello$ ls .git/refs
heads  tags
Jordan@Jordan:~/hello$ ls .git/refs/heads
master
Jordan@Jordan:~/hello$ ls .git/refs/tags
v1  v1-beta
Jordan@Jordan:~/hello$ cat .git/refs/tags/v1
b2fa0b0336f33e373591cd9314c571de0f413
Jordan@Jordan:~/hello$ cat .git/HEAD
ref: refs/heads/master
Jordan@Jordan:~/hello$
```

Рис. 3.18: Каталог .git

3.16 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. 3.19).

```
es Terminal anp 23 05:50
Jordan@jordan: ~/hello
Date: Tue Apr 23 05:35:51 2024 +0300
Added index.html.
Jordan@jordan:~/hello$ git cat-file -t d88616aad363bfadf39cf558c4c72355631f2668
commit
Jordan@jordan:~/hello$ git cat-file -p git cat-file -p <hash>
hash: syntax error near unexpected token `newline'
Jordan@jordan:~/hello$ git cat-file -p git cat-file -p d88616aad363bfadf39cf558c4c72355631f2668
usage: git cat-file [-t [<allow-unknown-type>] | -s [<allow-unknown-type>] | -e | -p | <type>] [--textconv] [--filters] [--path=<path>] <object>
or: git cat-file [--batch[=<format>] | --batch-check[=<format>]] [--follow-synlinks] [--textconv] [--filters]

<type> can be one of: blob, tree, commit, tag
-t show object type
-s show object size
-e exit with zero when there's no error
-p pretty-print object's content
--textconv for blob objects, run textconv on object's content
--filters for blob objects, run filters on object's content
--path <blob> use a specific path for --textconv/--filters
--allow-unknown-type allow -s and -t to work with broken/corrupt objects
--buffer buffer --batch output
--batch[=<format>] show info and content of objects fed from the standard input
--batch-check[=<format>] show info about objects fed from the standard input
--follow-synlinks follow in-tree synlinks (used with --batch or --batch-check)
--batch-all-objects show all objects with --batch or --batch-check
--unordered do not order --batch-all-objects output

Jordan@jordan:~/hello$ git cat-file -p d88616aad363bfadf39cf558c4c72355631f2668
tree 4d8fb158c2d4b97a629224a4ad98199acab1b83
parent a4dfec7d0b431f4ff62a57f225744bdb762cd2e5
author Jordaniakondzo <jordaniakondzo3438@gmail.com> 1713839751 +0300
committer Jordaniakondzo <jordaniakondzo3438@gmail.com> 1713839751 +0300

Added index.html.
Jordan@jordan:~/hello$ git cat-file -p bd8fb158c2d4b97a629224a4ad98199acab1b83
100644 blob 6a58ae4d328586a31568dfdf6f5ebba74230b5 index.html
040000 tree 3ab5e480ec562a88130aa003b1881b14725acc8a lib
Jordan@jordan:~/hello$ git cat-file -p 3ab5e480ec562a88130aa003b1881b14725acc8a
100644 blob c8dc0a091d74b719bffe7b004b7180e586c9f8 hello.html
Jordan@jordan:~/hello$ git cat-file -p c8dc0a091d74b719bffe7b004b7180e586c9f8
<!-- Author: Dmitry S. Kulyabov (kulyabov-ds@rudn.ru) -->
<html>
<head>
</head>
<body>
<h1>Hello, World!</h1>
</body>
```

Рис. 3.19: 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. 3.20).

```
es Terminal anp 23 06:05
Jordan@jordan: ~/hello
parent a4dfec7d0b431f4ff62a57f225744bdb762cd2e5
author Jordaniakondzo <jordaniakondzo3438@gmail.com> 1713839751 +0300
committer Jordaniakondzo <jordaniakondzo3438@gmail.com> 1713839751 +0300

Added index.html.
Jordan@jordan:~/hello$ git cat-file -p d88616aad363bfadf39cf558c4c72355631f2668
tree bd8fb158c2d4b97a629224a4ad98199acab1b83
parent a4dfec7d0b431f4ff62a57f225744bdb762cd2e5
author Jordaniakondzo <jordaniakondzo3438@gmail.com> 1713839751 +0300
committer Jordaniakondzo <jordaniakondzo3438@gmail.com> 1713839751 +0300

Added index.html.
Jordan@jordan:~/hello$ git cat-file -p a4dfec7d0b431f4ff62a57f225744bdb762cd2e5
tree 4ac0ee2b3f76a7f6d9eca37f5a947616ef15c4
parent 8909a5f9427a1b520b32a5367da5a4ec239d0324
author Jordaniakondzo <jordaniakondzo3438@gmail.com> 1713839529 +0300
committer Jordaniakondzo <jordaniakondzo3438@gmail.com> 1713839529 +0300

Moved hello.html to lib
Jordan@jordan:~/hello$ git cat-file -p 8909a5f9427a1b520b32a5367da5a4ec239d0324
tree 3ab5e480ec562a88130aa003b1881b14725acc8a
parent b2fa0b03336f33e373591c0d9314c571de0f413
author Jordaniakondzo <jordaniakondzo3438@gmail.com> 1713839112 +0300
committer Jordaniakondzo <jordaniakondzo3438@gmail.com> 1713839223 +0300

Add an author/email comment
Jordan@jordan:~/hello$ git cat-file -p b2fa0b03336f33e373591c0d9314c571de0f413
tree 4fd0ee7f742a5b372a80ca732ca0d4b158077c
parent ebf0d4b1ad859356256ccc14248c29842fbaee54
author Jordaniakondzo <jordaniakondzo3438@gmail.com> 1713836950 +0300
committer Jordaniakondzo <jordaniakondzo3438@gmail.com> 1713836950 +0300

Added HTML header
Jordan@jordan:~/hello$ git cat-file -p ebf0d4b1ad859356256ccc14248c29842fbaee54
tree bfe1d4f1469cea506fce19129492d8077224519
parent ff4bee7b9376a7f6d9eca37f5a947616ef15c4
author Jordaniakondzo <jordaniakondzo3438@gmail.com> 1713836844 +0300
committer Jordaniakondzo <jordaniakondzo3438@gmail.com> 1713836844 +0300

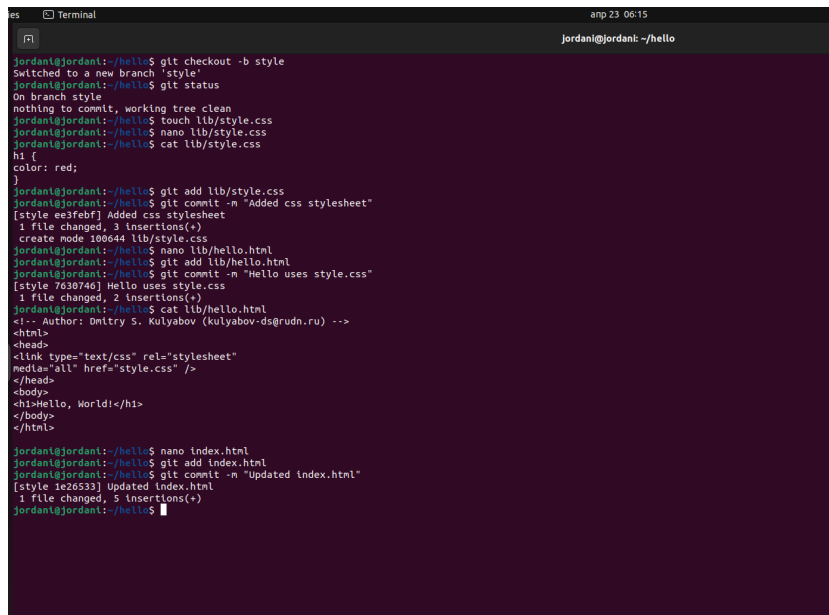
Added standard HTML page tags
Jordan@jordan:~/hello$ git cat-file -p ff4bee7b9376a7f6d9eca37f5a947616ef15c4
tree 629377ed8eb96afede9fdec37b481e4a1d3dc56
parent 39958c5c335173e87fceb1b5d4159139d4aae245
author Jordaniakondzo <jordaniakondzo3438@gmail.com> 1713836509 +0300
committer Jordaniakondzo <jordaniakondzo3438@gmail.com> 1713836509 +0300

adding
Jordan@jordan:~/hello$
```

Рис. 3.20: Поиск оригинального файла hello.html

3.17 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. 3.21).



```
JordanI@JordanI: ~/hello
JordanI@JordanI:~/hello$ git checkout -b style
Switched to a new branch 'style'
JordanI@JordanI:~/hello$ git status
On branch style
nothing to commit, working tree clean
JordanI@JordanI:~/hello$ touch lib/style.css
JordanI@JordanI:~/hello$ nano lib/style.css
JordanI@JordanI:~/hello$ cat lib/style.css
h1 {
  color: red;
}
JordanI@JordanI:~/hello$ git add lib/style.css
JordanI@JordanI:~/hello$ git commit -m "Added css stylesheet"
[style ee3feb9] Added css stylesheet
1 file changed, 3 insertions(+)
create mode 100644 lib/style.css
JordanI@JordanI:~/hello$ nano lib/hello.html
JordanI@JordanI:~/hello$ git add lib/hello.html
JordanI@JordanI:~/hello$ git commit -m "Hello uses style.css"
[style 7630746] Hello uses style.css
1 file changed, 2 insertions(+)
JordanI@JordanI:~/hello$ cat lib/hello.html
<!-- Author: Dmitry S. Kulyabov (kulyabov-ds@rudn.ru) -->
<html>
<head>
<link type="text/css" rel="stylesheet"
media="all" href="style.css" />
</head>
<body>
<h1>Hello, World!</h1>
</body>
</html>
JordanI@JordanI:~/hello$ nano index.html
JordanI@JordanI:~/hello$ git add index.html
JordanI@JordanI:~/hello$ git commit -m "Updated index.html"
[style 1e26533] Updated index.html
1 file changed, 5 insertions(+)
JordanI@JordanI:~/hello$
```

Рис. 3.21: Создание ветки

3.18 Навигация по веткам

Посмотрим все логи(рис. fig. 3.22).

```
es Terminal amp 23 06:18
Jordani@Jordani: ~/hello

Jordani@Jordani:~/hello$ git log --all
commit 1e265330fd886604665d8e1a85b4e18cfb2c9e1a (HEAD -> style)
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 06:15:39 2024 +0300

    Updated index.html

commit 76387464c98731342276fcd869e805d1ef88
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 06:13:16 2024 +0300

    Hello uses style.css

commit ee3febfff888c5d962144bee1f8d7cc0834a23d0
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 06:09:23 2024 +0300

    Added css stylesheet

commit d88616aad363bfad39c9f550c4c72355631f2660 (master)
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 05:35:51 2024 +0300

    Added index.html.

commit addfec7d0b431f4ff02a57f225744bdb762cd2e5
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 05:32:09 2024 +0300

    Moved hello.html to lib

commit 8909a5f9427a1b520b32a5367da5a4ec239d0324
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 05:25:12 2024 +0300

    Add an author/email comment

commit b2fa0b0b336f39e373591c0d9314c571de0f413 (tag: v1)
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:49:10 2024 +0300

    Added HTML header

commit ebf0d4b1ad859356256cccc14248c29842fbaee54 (tag: v1-beta)
Author: Jordaniakondzo <jordaniakondzo3438@gmail.com>
Date: Tue Apr 23 04:47:24 2024 +0300

    Added standard HTML page tags
```

Рис. 3.22: Просмотр логов новой ветки

Переключимся обратно на основную ветку и посмотрим содержимое файла `lib/hello.html`, заметим, что он не использует стили, также посмотрим содержимое этого файла в новой ветке

4 Выводы

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