

# **Лабораторная работа №1**

**Работа с git**

Галацан Николай, НПИбд-01-22

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>6</b>
<b>2</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
3.1	Подготовка . . . . .	8
3.2	Создание проекта . . . . .	8
3.3	Внесение и индексация изменений . . . . .	9
3.4	История . . . . .	11
3.5	Получение старых версий . . . . .	12
3.6	Создание тегов версий . . . . .	13
3.7	Отмена локальных изменений (до индексации) . . . . .	15
3.8	Отмена проиндексированных изменений (перед коммитом) . . . . .	16
3.9	Отмена коммитов . . . . .	17
3.10	Удаление коммитов из ветки . . . . .	18
3.11	Удаление тега оорс . . . . .	19
3.12	Внесение изменений в коммиты . . . . .	20
3.13	Перемещение файлов . . . . .	21
3.14	Git внутри: Каталог .git . . . . .	22
3.15	Создание ветки . . . . .	25
3.16	Изменения в ветке master . . . . .	27
3.17	Слияние . . . . .	27
3.18	Создание конфликта . . . . .	28
3.19	Сброс ветки style . . . . .	29
3.20	Сброс ветки master . . . . .	30
3.21	Перебазирование . . . . .	31
3.22	Слияние в ветку master . . . . .	31
3.23	Клонирование репозитория . . . . .	32
3.24	Удаленные ветки . . . . .	33
3.25	Изменение оригинального репозитория . . . . .	33
3.26	Извлечение изменений . . . . .	33
3.27	Слияние извлеченных изменений . . . . .	34
3.28	Добавление ветки наблюдения . . . . .	34
3.29	Чистые репозитории . . . . .	35
3.30	Извлечение общих изменений . . . . .	36
<b>4</b>	<b>Выводы</b>	<b>37</b>



# Список иллюстраций

3.1	Настройка основных параметров git . . . . .	8
3.2	Создание рабочего пространства и репозитория . . . . .	9
3.3	Внесение изменений и индексация, проверка статуса . . . . .	9
3.4	Коммит первоначальных изменений . . . . .	10
3.5	Добавление стандартных тегов . . . . .	10
3.6	Внесение изменений, индексирование, коммит . . . . .	11
3.7	Просмотр истории . . . . .	12
3.8	Получение старых версий . . . . .	13
3.9	Создание тега . . . . .	13
3.10	Переключение по тегам и просмотр . . . . .	14
3.11	Просмотр тегов в логе . . . . .	14
3.12	Отмена изменений до индексации . . . . .	15
3.13	Отмена изменений после индексации . . . . .	16
3.14	Отмена изменений после коммита . . . . .	17
3.15	Нежелательные коммиты . . . . .	18
3.16	Сброс коммитов . . . . .	19
3.17	Удаление тега и коммитов . . . . .	20
3.18	Изменение коммита . . . . .	21
3.19	Перемещение файла . . . . .	22
3.20	index.html . . . . .	22
3.21	index.html в браузере . . . . .	22
3.22	Просмотр .git . . . . .	23
3.23	Просмотр по хэшу . . . . .	24
3.24	Поиск первоначального файла hello.html по хэшу . . . . .	25
3.25	Создание новой ветки и изменение файлов . . . . .	26
3.26	Просмотр файлов на двух ветках . . . . .	26
3.27	Создание README.md в master . . . . .	27
3.28	Просмотр веток . . . . .	27
3.29	Слияние master с style . . . . .	28
3.30	Создание конфликта . . . . .	29
3.31	Разрешение конфликта и слияние . . . . .	29
3.32	Сброс ветки style по хэшу . . . . .	30
3.33	Сброс ветки master по хэшу . . . . .	30
3.34	Перебазирование . . . . .	31
3.35	Слияние style в master . . . . .	32
3.36	Создание клона репозитория, просмотр файлов и коммитов . . . .	32
3.37	Просмотр веток в клоне . . . . .	33

3.38 Извлечение изменений из оригинального репозитория . . . . .	34
3.39 Слияние извлеченных изменений . . . . .	34
3.40 Добавление ветки наблюдения . . . . .	35
3.41 Создание чистого репозитория, добавление в оригинальный, от- правка изменений в общий . . . . .	35
3.42 Извлечение общих изменений, проверка . . . . .	36

# 1 Цель работы

Научиться использовать систему управления версиями git, освоить основные команды, применить на практике полученные знания.

## 2 Теоретическое введение

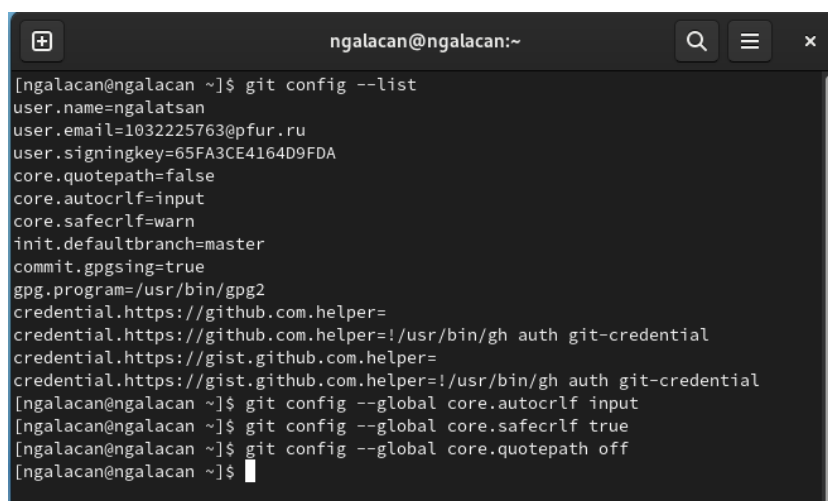
Git — распределённая система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года; координатор — Дзюн Хамано. Среди проектов, использующих Git, — ядро Linux, Swift, Android, Drupal, Cairo, GNU Core Utilities, Mesa, Wine, Chromium, Compiz Fusion, FlightGear, jQuery, PHP, NASM, MediaWiki, DokuWiki, Qt, ряд дистрибутивов Linux.

Программа является свободной и выпущена под лицензией GNU GPL версии 2. По умолчанию используется TCP-порт 9418 [1].

## 3 Выполнение лабораторной работы

### 3.1 Подготовка

Сначала проверяю имя пользователя и электронную почту. Приступаю к настройке `core.autocrlf`, `core.safecrlf`, `core.quoteopath` (рис. 3.1).



```
ngalacan@ngalacan:~  
[ngalacan@ngalacan ~]$ git config --list  
user.name=ngalatsan  
user.email=1032225763@pfur.ru  
user.signingkey=65FA3CE4164D9FDA  
core.quoteopath=false  
core.autocrlf=input  
core.safecrlf=warn  
init.defaultbranch=master  
commit.gpgsign=true  
gpg.program=/usr/bin/gpg2  
credential.https://github.com.helper=  
credential.https://github.com.helper=!/usr/bin/gh auth git-credential  
credential.https://gist.github.com.helper=  
credential.https://gist.github.com.helper=!/usr/bin/gh auth git-credential  
[ngalacan@ngalacan ~]$ git config --global core.autocrlf input  
[ngalacan@ngalacan ~]$ git config --global core.safecrlf true  
[ngalacan@ngalacan ~]$ git config --global core.quoteopath off  
[ngalacan@ngalacan ~]$
```

Рис. 3.1: Настройка основных параметров git

### 3.2 Создание проекта

Создаю рабочий каталог с файлом `hello.html`. Добавляю запись. Создаю репозиторий и добавляю файл, проверяю текущее состояние. В репозитории хранится текущее состояние рабочего каталога, и нет никаких изменений, ожидающих записи (рис. 3.2).



```

[ngalacan@ngalacan ~]$ mkdir hello
[ngalacan@ngalacan ~]$ cd hello
[ngalacan@ngalacan hello]$ touch hello.html
[ngalacan@ngalacan hello]$ echo "Hello, world!" > hello.html
[ngalacan@ngalacan hello]$ git init
Инициализирован пустой репозиторий Git в /home/ngalacan/hello/.git/
[ngalacan@ngalacan hello]$ git add hello.html
[ngalacan@ngalacan hello]$ git commit -m 'Initial commit'
[master (корневой коммит) b5d5965] Initial commit
1 file changed, 1 insertion(+)
create mode 100644 hello.html
[ngalacan@ngalacan hello]$ git status
Текущая ветка: master
нечего коммитить, нет изменений в рабочем каталоге
[ngalacan@ngalacan hello]$

```

Рис. 3.2: Создание рабочего пространства и репозитория

### 3.3 Внесение и индексация изменений

Изменяю содержимое файла на `<h1>Hello, World!</h1>`. Проверяю состояние (git знает, что файл `hello.html` был изменен, но при этом эти изменения еще не зафиксированы в репозитории), индексирую изменения и вновь проверяю (изменения файла `hello.html` были проиндексированы. Это означает, что git теперь знает об изменении, но изменение пока не записано в репозиторий) (рис. 3.3).

```

[ngalacan@ngalacan hello]$ gedit hello.html
[ngalacan@ngalacan hello]$ git status
Текущая ветка: master
Изменения, которые не в индексе для коммита:
(используйте «git add <файл>...», чтобы добавить файл в индекс)
(используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)

изменено:      hello.html

индекс пуст (используйте «git add» и/или «git commit -a»)
[ngalacan@ngalacan hello]$ git add hello.html
[ngalacan@ngalacan hello]$ git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
(используйте «git restore --staged <файл>...», чтобы убрать из индекса)

изменено:      hello.html
[ngalacan@ngalacan hello]$

```

Рис. 3.3: Внесение изменений и индексация, проверка статуса

Делаю коммит изменений (рис. 3.4).

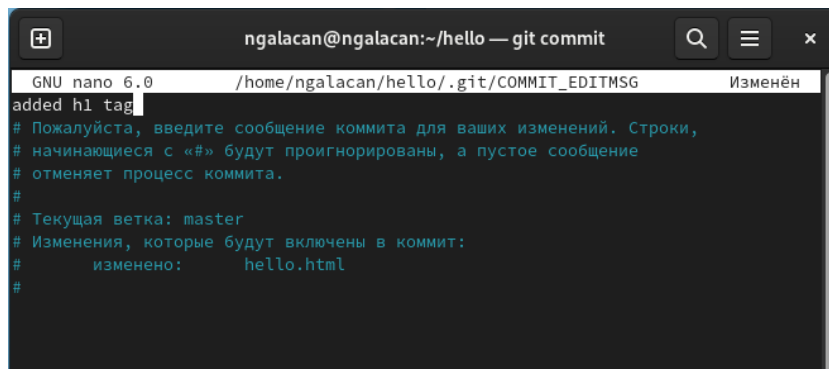


Рис. 3.4: Коммит первоначальных изменений

Еще раз проверив статус, убеждаюсь, что рабочий каталог чистый. Изменяю страницу «Hello, World!», чтобы она содержала стандартные теги (рис. 3.5).

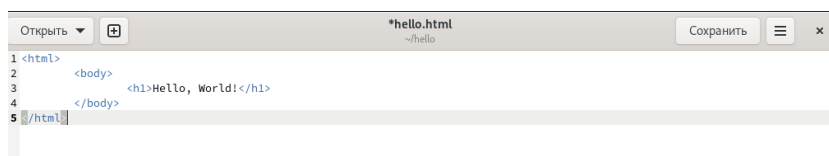
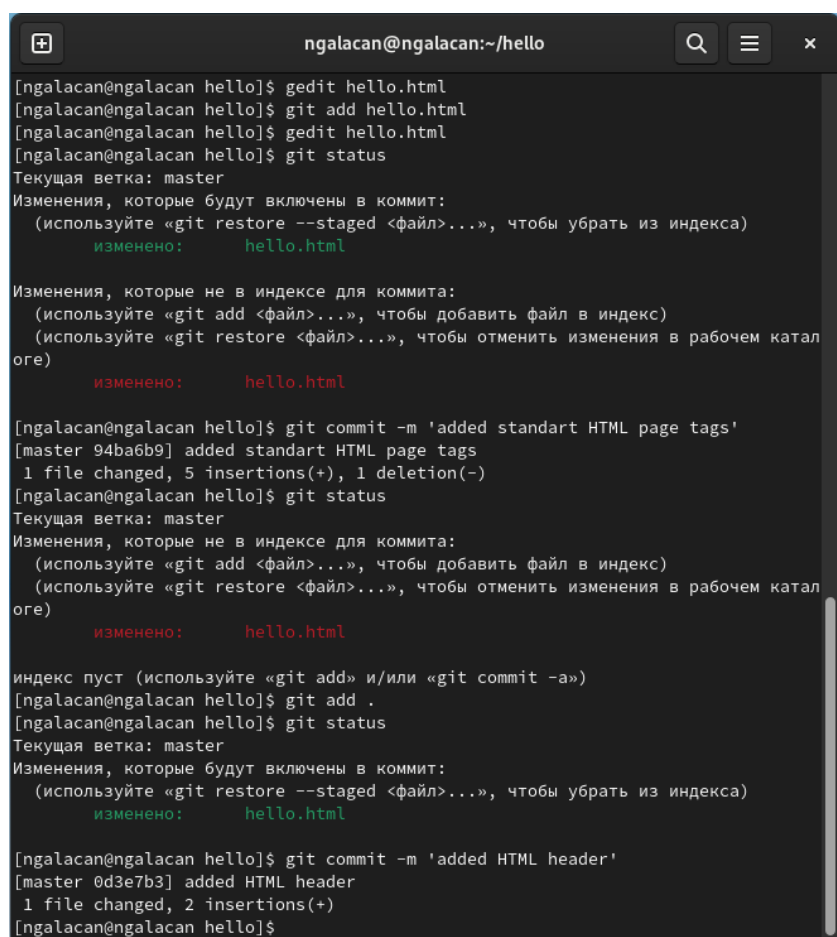


Рис. 3.5: Добавление стандартных тегов

Теперь добавляю это изменение в индекс git. В hello.html добавляю секцию <head>:

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

Проверяю статус: hello.html указан дважды в состоянии. Делаю коммит индексированного изменения, просматриваю статус: осталось только непроиндексированное изменение. Индексирую и проверяю. Делаю еще один коммит (рис. 3.6).



```
[ngalacan@ngalacan hello]$ gedit hello.html
[ngalacan@ngalacan hello]$ git add hello.html
[ngalacan@ngalacan hello]$ gedit hello.html
[ngalacan@ngalacan hello]$ git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
        изменено:      hello.html

Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>...», чтобы добавить файл в индекс)
  (используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)
        изменено:      hello.html

[ngalacan@ngalacan hello]$ git commit -m 'added standart HTML page tags'
[master 94ba6b9] added standart HTML page tags
 1 file changed, 5 insertions(+), 1 deletion(-)
[ngalacan@ngalacan hello]$ git status
Текущая ветка: master
Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>...», чтобы добавить файл в индекс)
  (используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)
        изменено:      hello.html

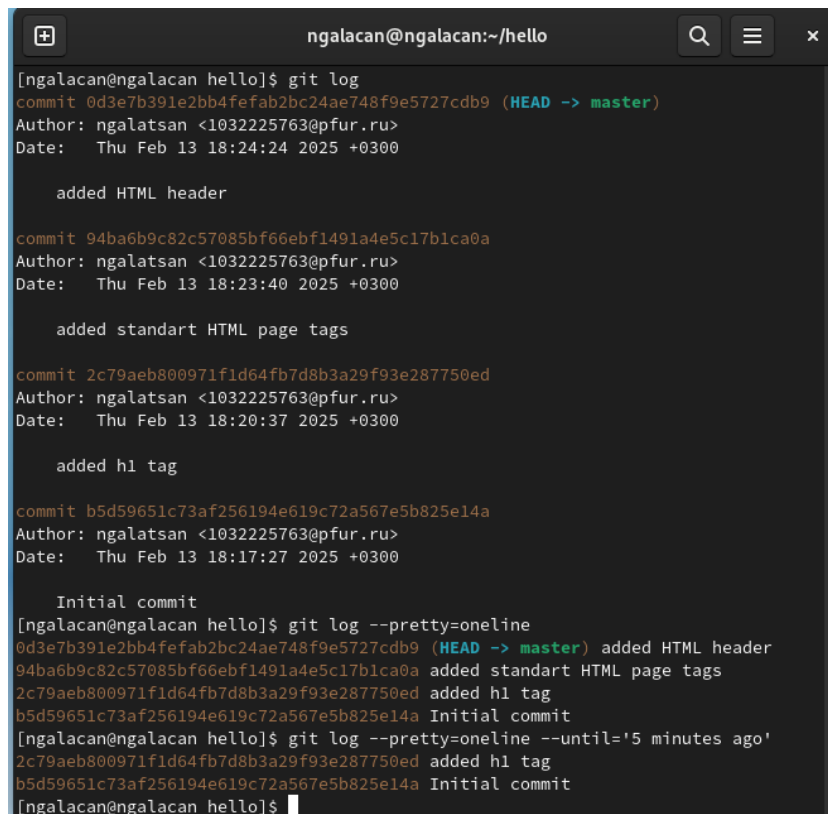
индекс пуст (используйте «git add» и/или «git commit -a»)
[ngalacan@ngalacan hello]$ git add .
[ngalacan@ngalacan hello]$ git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
        изменено:      hello.html

[ngalacan@ngalacan hello]$ git commit -m 'added HTML header'
[master 0d3e7b3] added HTML header
 1 file changed, 2 insertions(+)
```

Рис. 3.6: Внесение изменений, индексирование, коммит

## 3.4 История

Получаю историю изменений. Просматриваю в разных форматах (рис. 3.7).

A terminal window titled 'ngalacan@ngalacan:~/hello' with search, menu, and close icons in the title bar. The terminal shows the output of 'git log' and 'git log --pretty=oneline'. The first 'git log' command shows four commits in detail, including commit hashes, authors, dates, and descriptions. The second 'git log --pretty=oneline' command shows the same four commits in a more compact format. The third 'git log --pretty=oneline --until='5 minutes ago'' command shows the last two commits from the previous list.

```
[ngalacan@ngalacan hello]$ git log
commit 0d3e7b391e2bb4fefab2bc24ae748f9e5727cdb9 (HEAD -> master)
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:24:24 2025 +0300

    added HTML header

commit 94ba6b9c82c57085bf66ebf1491a4e5c17b1ca0a
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:23:40 2025 +0300

    added standart HTML page tags

commit 2c79aeb800971f1d64fb7d8b3a29f93e287750ed
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:20:37 2025 +0300

    added h1 tag

commit b5d59651c73af256194e619c72a567e5b825e14a
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:17:27 2025 +0300

    Initial commit

[ngalacan@ngalacan hello]$ git log --pretty=oneline
0d3e7b391e2bb4fefab2bc24ae748f9e5727cdb9 (HEAD -> master) added HTML header
94ba6b9c82c57085bf66ebf1491a4e5c17b1ca0a added standart HTML page tags
2c79aeb800971f1d64fb7d8b3a29f93e287750ed added h1 tag
b5d59651c73af256194e619c72a567e5b825e14a Initial commit

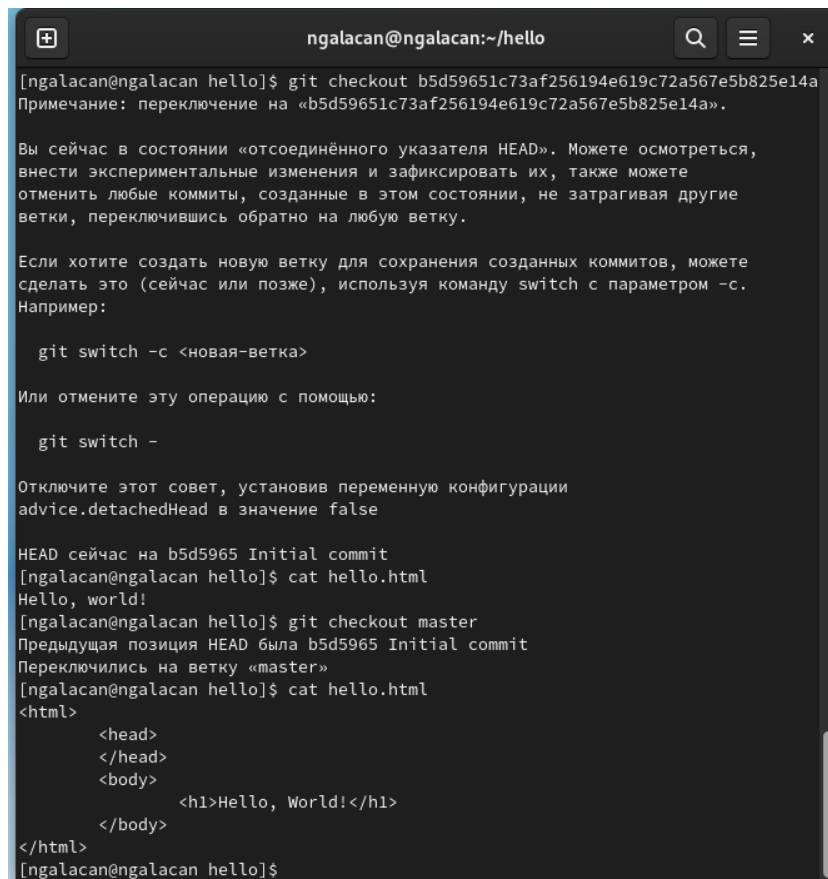
[ngalacan@ngalacan hello]$ git log --pretty=oneline --until='5 minutes ago'
2c79aeb800971f1d64fb7d8b3a29f93e287750ed added h1 tag
b5d59651c73af256194e619c72a567e5b825e14a Initial commit

[ngalacan@ngalacan hello]$
```

Рис. 3.7: Просмотр истории

## 3.5 Получение старых версий

Изучаю данные лога и нахожу хэш для первого коммита. Использую этот хэш-код для получения старой версии. Возвращаюсь к последней версии (рис. 3.8).



```
ngalacan@ngalacan:~/hello
[ngalacan@ngalacan hello]$ git checkout b5d59651c73af256194e619c72a567e5b825e14a
Примечание: переключение на «b5d59651c73af256194e619c72a567e5b825e14a».

Вы сейчас в состоянии «отсоединённого указателя HEAD». Можете осмотреться,
внести экспериментальные изменения и зафиксировать их, также можете
отменить любые коммиты, созданные в этом состоянии, не затрагивая другие
ветки, переключившись обратно на любую ветку.

Если хотите создать новую ветку для сохранения созданных коммитов, можете
сделать это (сейчас или позже), используя команду switch с параметром -с.
Например:

    git switch -с <новая-ветка>

Или отмените эту операцию с помощью:

    git switch -

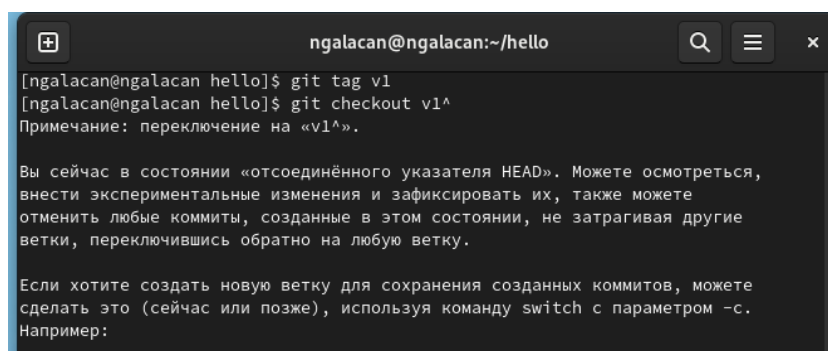
Отключите этот совет, установив переменную конфигурации
advice.detachedHead в значение false

HEAD сейчас на b5d5965 Initial commit
[ngalacan@ngalacan hello]$ cat hello.html
Hello, world!
[ngalacan@ngalacan hello]$ git checkout master
Предыдущая позиция HEAD была b5d5965 Initial commit
Переключились на ветку «master»
[ngalacan@ngalacan hello]$ cat hello.html
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
[ngalacan@ngalacan hello]$
```

Рис. 3.8: Получение старых версий

## 3.6 Создание тегов версий

Создаю тег текущей версии (рис. 3.9), предыдущей версии и переключаюсь между ними, просматриваю теги (рис. 3.10)



```
ngalacan@ngalacan:~/hello
[ngalacan@ngalacan hello]$ git tag v1
[ngalacan@ngalacan hello]$ git checkout v1^
Примечание: переключение на «v1^».

Вы сейчас в состоянии «отсоединённого указателя HEAD». Можете осмотреться,
внести экспериментальные изменения и зафиксировать их, также можете
отменить любые коммиты, созданные в этом состоянии, не затрагивая другие
ветки, переключившись обратно на любую ветку.

Если хотите создать новую ветку для сохранения созданных коммитов, можете
сделать это (сейчас или позже), используя команду switch с параметром -с.
Например:
```

Рис. 3.9: Создание тега

```
ngalacan@ngalacan:~/hello
[ngalacan@ngalacan hello]$ cat hello.html
<html>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
[ngalacan@ngalacan hello]$ git tag v1-beta
[ngalacan@ngalacan hello]$ git checkout v1
Предыдущая позиция HEAD была 94ba6b9 added standart HTML page tags
HEAD сейчас на 0d3e7b3 added HTML header
[ngalacan@ngalacan hello]$ git checkout v1-beta
Предыдущая позиция HEAD была 0d3e7b3 added HTML header
HEAD сейчас на 94ba6b9 added standart HTML page tags
[ngalacan@ngalacan hello]$ git tag
v1
v1-beta
```

Рис. 3.10: Переключение по тегам и просмотр

Просматриваю теги в логе. Вижу теги (v1 и v1-beta) в логе вместе с именем ветки (master). Кроме того HEAD показывает коммит, на который я переключился (на данный момент это v1-beta) (рис. 3.11).

```
ngalacan@ngalacan:~/hello
[ngalacan@ngalacan hello]$ git log master -al
error: switch `l' requires a value
[ngalacan@ngalacan hello]$ git log master --all
commit 0d3e7b391e2bb4fefab2bc24ae748f9e5727cdb9 (tag: v1, master)
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:24:24 2025 +0300

    added HTML header

commit 94ba6b9c82c57085bf66ebf1491a4e5c17b1ca0a (HEAD, tag: v1-beta)
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:23:40 2025 +0300

    added standart HTML page tags

commit 2c79aeb800971f1d64fb7d8b3a29f93e287750ed
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:20:37 2025 +0300

    added h1 tag

commit b5d59651c73af256194e619c72a567e5b825e14a
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:17:27 2025 +0300

    Initial commit
[ngalacan@ngalacan hello]$
```

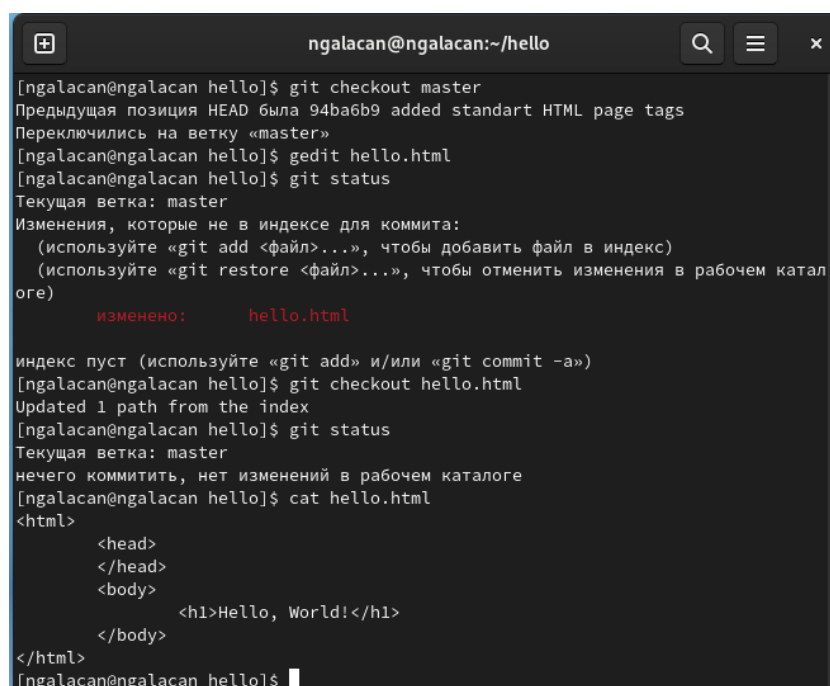
Рис. 3.11: Просмотр тегов в логе

## 3.7 Отмена локальных изменений (до индексации)

Переключившись на ветку master изменяю файл:

```
<html>
  <head>
</head>
  <body>
    <h1>Hello, World!</h1>
    <!-- This is a bad comment. We want to revert it. -->
  </body>
</html>
```

Проверяю состояние: файл hello.html был изменен, но еще не проиндексирован. Для переключения версии файла использую команды, проверяю статус, просматриваю файл (рис. 3.12).



```
ngalacan@ngalacan:~/hello
[ngalacan@ngalacan hello]$ git checkout master
Предыдущая позиция HEAD была 94ba6b9 added standart HTML page tags
Переключились на ветку «master»
[ngalacan@ngalacan hello]$ gedit hello.html
[ngalacan@ngalacan hello]$ git status
Текущая ветка: master
Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>...», чтобы добавить файл в индекс)
  (используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)

изменено:      hello.html

индекс пуст (используйте «git add» и/или «git commit -a»)
[ngalacan@ngalacan hello]$ git checkout hello.html
Updated 1 path from the index
[ngalacan@ngalacan hello]$ git status
Текущая ветка: master
нечего коммитить, нет изменений в рабочем каталоге
[ngalacan@ngalacan hello]$ cat hello.html
<html>
  <head>
</head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
[ngalacan@ngalacan hello]$
```

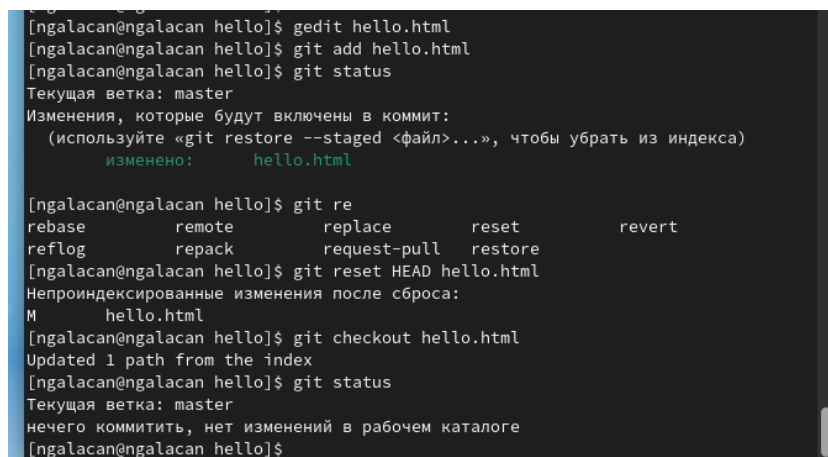
Рис. 3.12: Отмена изменений до индексации

## 3.8 Отмена проиндексированных изменений (перед коммитом)

Вновь вношу изменений в файл:

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

Индексирую изменения. Выполняю сброс буферной зоны. Команда `git reset` сбрасывает буферную зону к HEAD. Это очищает буферную зону от изменений, которые мы только что проиндексировали. Использую команду `git checkout`, чтобы удалить нежелательные изменения в рабочем каталоге (рис. 3.13).



```
[ngalacan@ngalacan hello]$ gedit hello.html
[ngalacan@ngalacan hello]$ git add hello.html
[ngalacan@ngalacan hello]$ git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    изменено:   hello.html

[ngalacan@ngalacan hello]$ git re
rebase      remote      replace     reset       revert
reflog      repack      request-pull restore
[ngalacan@ngalacan hello]$ git reset HEAD hello.html
Непроиндексированные изменения после сброса:
M       hello.html
[ngalacan@ngalacan hello]$ git checkout hello.html
Updated 1 path from the index
[ngalacan@ngalacan hello]$ git status
Текущая ветка: master
ничего коммитить, нет изменений в рабочем каталоге
[ngalacan@ngalacan hello]$
```

Рис. 3.13: Отмена изменений после индексации

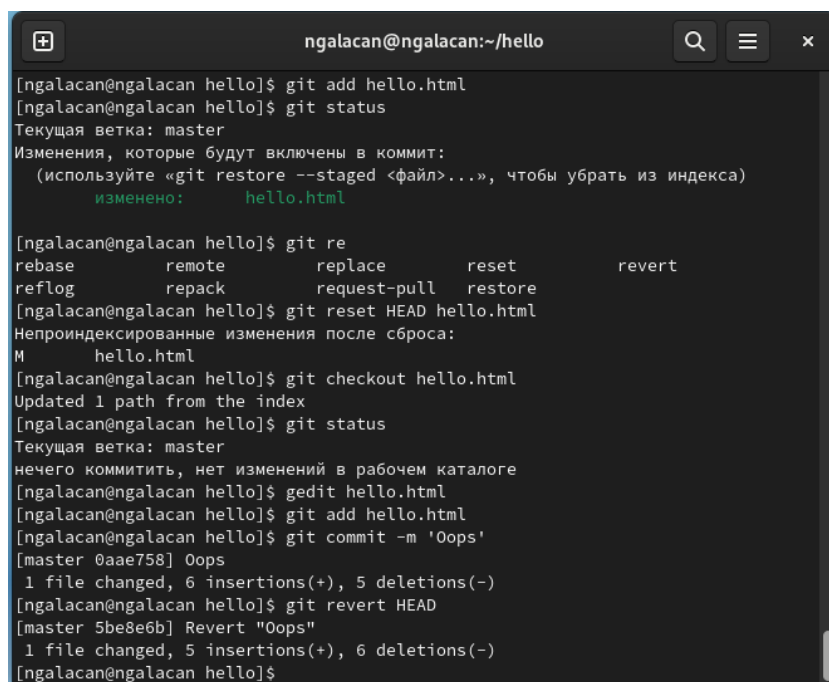


## 3.9 Отмена коммитов

Изменяю файл:

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

Индексирую и коммичу изменение. Чтобы отменить коммит, необходимо сделать коммит, который удаляет изменения, сохраненные нежелательным коммитом (рис. 3.14).

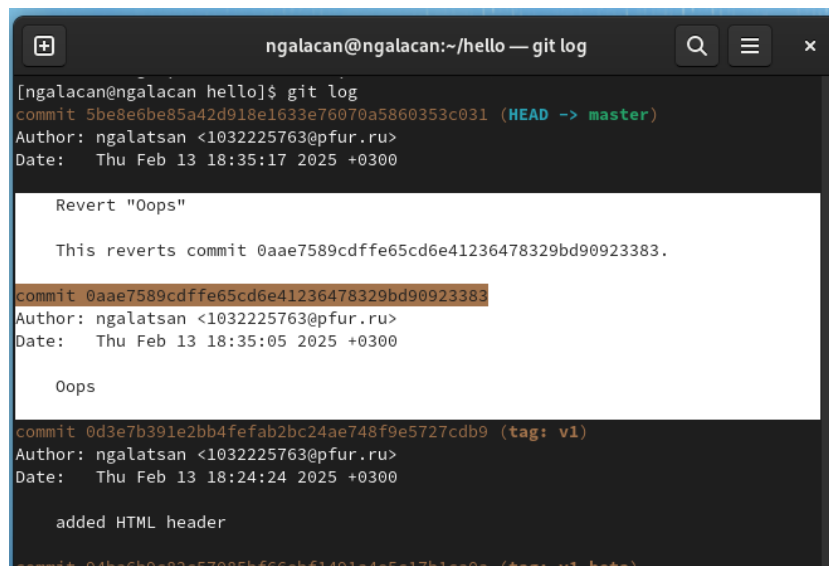


```
ngalacan@ngalacan:~/hello
[ngalacan@ngalacan hello]$ git add hello.html
[ngalacan@ngalacan hello]$ git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    изменено:   hello.html

[ngalacan@ngalacan hello]$ git re
rebase      remote      replace     reset       revert
reflog      repack      request-pull restore
[ngalacan@ngalacan hello]$ git reset HEAD hello.html
Неприндексированные изменения после сброса:
M       hello.html
[ngalacan@ngalacan hello]$ git checkout hello.html
Updated 1 path from the index
[ngalacan@ngalacan hello]$ git status
Текущая ветка: master
нечего коммитить, нет изменений в рабочем каталоге
[ngalacan@ngalacan hello]$ gedit hello.html
[ngalacan@ngalacan hello]$ git add hello.html
[ngalacan@ngalacan hello]$ git commit -m 'Oops'
[master 0aae758] Oops
 1 file changed, 6 insertions(+), 5 deletions(-)
[ngalacan@ngalacan hello]$ git revert HEAD
[master 5be8e6b] Revert "Oops"
 1 file changed, 5 insertions(+), 6 deletions(-)
[ngalacan@ngalacan hello]$
```

Рис. 3.14: Отмена изменений после коммита

Проверив лог, вижу оба коммита (рис. 3.15).



```
ngalacan@ngalacan:~/hello — git log
[ngalacan@ngalacan hello]$ git log
commit 5be8e6be85a42d918e1633e76070a5860353c031 (HEAD -> master)
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:35:17 2025 +0300

    Revert "Oops"

    This reverts commit 0aae7589cdffe65cd6e41236478329bd90923383.

commit 0aae7589cdffe65cd6e41236478329bd90923383
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:35:05 2025 +0300

    Oops

commit 0d3e7b391e2bb4fefab2bc24ae748f9e5727cdb9 (tag: v1)
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:24:24 2025 +0300

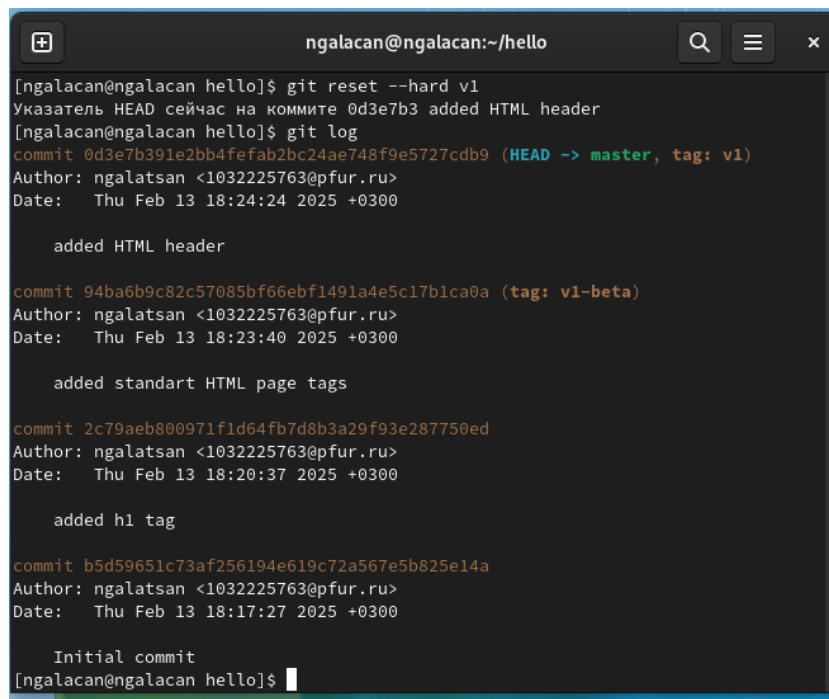
    added HTML header

commit 94ba5b9c82c57085bf66ebf1491a4e5c17b1ca9a (tag: v1-beta)
```

Рис. 3.15: Нежелательные коммиты

## 3.10 Удаление коммитов из ветки

Удаляю последние два коммита с помощью сброса, сначала отметив последний коммит тегом `oops`, чтобы его можно было потом найти. Использую 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 версии (рис. 3.16).

A terminal window titled 'ngalacan@ngalacan:~/hello' with search, menu, and close icons. It shows the execution of 'git reset --hard v1' followed by 'git log'. The log output shows a sequence of commits: an initial commit, a commit adding an h1 tag, a commit adding standard HTML page tags, a commit adding HTML header, and a commit adding a tag 'v1-beta'. The HEAD is now at 'master' pointing to commit 'v1'.

```
[ngalacan@ngalacan hello]$ git reset --hard v1
Указатель HEAD сейчас на коммите 0d3e7b3 added HTML header
[ngalacan@ngalacan hello]$ git log
commit 0d3e7b391e2bb4fefab2bc24ae748f9e5727cdb9 (HEAD -> master, tag: v1)
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:24:24 2025 +0300

    added HTML header

commit 94ba6b9c82c57085bf66ebf1491a4e5c17b1ca0a (tag: v1-beta)
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:23:40 2025 +0300

    added standart HTML page tags

commit 2c79aeb800971f1d64fb7d8b3a29f93e287750ed
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:20:37 2025 +0300

    added h1 tag

commit b5d59651c73af256194e619c72a567e5b825e14a
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:17:27 2025 +0300

    Initial commit
[ngalacan@ngalacan hello]$
```

Рис. 3.16: Сброс коммитов

## 3.11 Удаление тега оорс

Удаляю тег и коммиты, на которые он ссылался, сборщиком мусора (рис. 3.17).

```
ngalacan@ngalacan:~/hello
[ngalacan@ngalacan hello]$ git tag -d oops
Метка «oops» удалена (была 5be8e6b)
[ngalacan@ngalacan hello]$ git log --all
commit 0d3e7b391e2bb4fefab2bc24ae748f9e5727cdb9 (HEAD -> master, tag: v1)
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:24:24 2025 +0300

    added HTML header

commit 94ba6b9c82c57085bf66ebf1491a4e5c17b1ca0a (tag: v1-beta)
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:23:40 2025 +0300

    added standart HTML page tags

commit 2c79aeb800971f1d64fb7d8b3a29f93e287750ed
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:20:37 2025 +0300

    added h1 tag

commit b5d59651c73af256194e619c72a567e5b825e14a
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:17:27 2025 +0300

    Initial commit
[ngalacan@ngalacan hello]$
```

Рис. 3.17: Удаление тега и коммитов

## 3.12 Внесение изменений в коммиты

Изменяю файл:

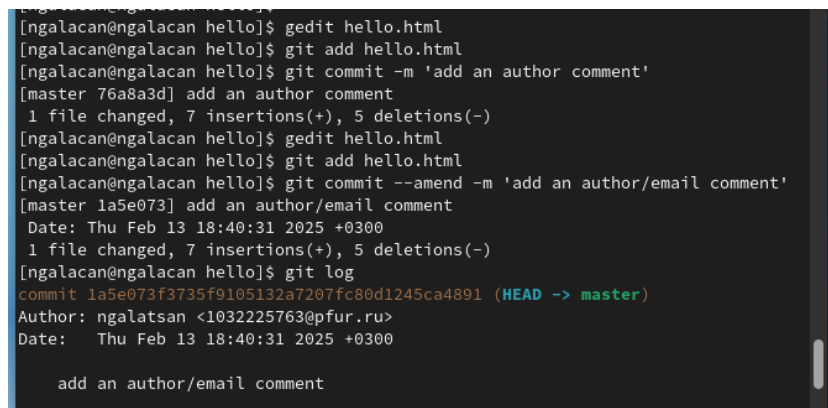
```
<!-- Author: N. Galatsan -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

Индексирую изменения и делаю коммит. Добавляю в файл информацию об электронной почте:

```
<!-- Author: N. Galatsan (1032225763@pfur.ru) -->
```

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

Изменяю предыдущий коммит. Можно 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 (рис. 3.18).



```
[ngalacan@ngalacan hello]$ gedit hello.html
[ngalacan@ngalacan hello]$ git add hello.html
[ngalacan@ngalacan hello]$ git commit -m 'add an author comment'
[master 76a8a3d] add an author comment
1 file changed, 7 insertions(+), 5 deletions(-)
[ngalacan@ngalacan hello]$ gedit hello.html
[ngalacan@ngalacan hello]$ git add hello.html
[ngalacan@ngalacan hello]$ git commit --amend -m 'add an author/email comment'
[master 1a5e073] add an author/email comment
Date: Thu Feb 13 18:40:31 2025 +0300
1 file changed, 7 insertions(+), 5 deletions(-)
[ngalacan@ngalacan hello]$ git log
commit 1a5e073f3735f9105132a7207fc80d1245ca4891 (HEAD -> master)
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:40:31 2025 +0300

    add an author/email comment
```

Рис. 3.18: Изменение коммита

## 3.13 Перемещение файлов

Создаю каталог `lib` и перемещаю туда файл `hello.html` с помощью `git mv`. Делаю коммит, добавляю файл `index.html` (рис. 3.20) в репозиторий, делаю коммит (рис. 3.19).

```

[ngalacan@ngalacan hello]$ mkdir lib
[ngalacan@ngalacan hello]$ git mv hello.html lib
[ngalacan@ngalacan hello]$ git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
(используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    переименовано: hello.html -> lib/hello.html

[ngalacan@ngalacan hello]$ git commit -m 'moved hello.html to lib'
[master ad56cb5] moved hello.html to lib
1 file changed, 0 insertions(+), 0 deletions(-)
rename hello.html => lib/hello.html (100%)
[ngalacan@ngalacan hello]$ touch index.html
[ngalacan@ngalacan hello]$ gedit index.html
[ngalacan@ngalacan hello]$

```

Рис. 3.19: Перемещение файла



```

*index.html
~/hello
1 <html>
2 <body>
3 <iframe src="lib/hello.html" width="200" height="200" />
4 </body>
5 </html>

```

Рис. 3.20: index.html

Теперь при открытии `index.html`, можно увидеть кусок страницы `hello` в маленьком окошке (рис. 3.21).



Рис. 3.21: index.html в браузере

### 3.14 Git внутри: Каталог `.git`

Просматриваю данные из каталога `.git`, объектные файлы, ветки, теги и хэш (рис. 3.22).

```

Create mode 100044 index.html
[ngalacan@ngalacan hello]$ ls -C .git
branches      config        HEAD  index  logs      ORIG_HEAD  refs
COMMIT_EDITMSG  description  hooks  info   objects  packed-refs
[ngalacan@ngalacan hello]$ ls -C .git/objects
00  0d  2c  5b  6d  7c  93  9f  aa  af  b7  cd  da  f9  info
0a  1a  5a  68  76  81  94  a4  ad  b5  bd  d0  dd  fe  pack
[ngalacan@ngalacan hello]$ ls -C .git/objects/5a
08fc805109b6b3add3ef11bda23005ce76f6b8
[ngalacan@ngalacan hello]$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
[ngalacan@ngalacan hello]$ ls .git/refs
heads  tags
[ngalacan@ngalacan hello]$ ls .git/refs/heads
master
[ngalacan@ngalacan hello]$ ls .git/refs/tags
v1  v1-beta
[ngalacan@ngalacan hello]$ cat .git/refs/tags/v1
0d3e7b391e2bb4fefab2bc24ae748f9e5727cdb9
[ngalacan@ngalacan hello]$

```

Рис. 3.22: Просмотр .git

Просматриваю файл HEAD, который содержит ссылку на текущую ветку. Ищу последний коммит. Вывожу последний коммит с помощью SHA1 хэша, просматриваю дерево, каталог lib, файл hello.html (рис. 3.23).

```
ngalacan@ngalacan:~/hello
[ngalacan@ngalacan hello]$ cat .git/HEAD
ref: refs/heads/master
[ngalacan@ngalacan hello]$ git log --max-count=1
commit fe917edbd342a6e3957d144ab8c0c01c20c6bb18 (HEAD -> master)
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:44:29 2025 +0300

    added index.html
[ngalacan@ngalacan hello]$ git cat-file -t fe917edbd342a6e3957d144ab8c0c01c20c6bb18
commit
[ngalacan@ngalacan hello]$ git cat-file -p fe917edbd342a6e3957d144ab8c0c01c20c6bb18
tree dd7af4f7618057c3ae38064891b13c937a5c6332
parent ad56cb54ebf9dd1da406af27d517d124a49ac96c
author ngalatsan <1032225763@pfur.ru> 1739461469 +0300
committer ngalatsan <1032225763@pfur.ru> 1739461469 +0300

added index.html
[ngalacan@ngalacan hello]$ git cat-file -p dd7af4f7618057c3ae38064891b13c937a5c6332
100644 blob 68583eacd328506a31568f0df6f45ebb874238b5    index.html
040000 tree 6d88b056751e0c2b9e008ce9eaf4a5992c559ae3    lib
[ngalacan@ngalacan hello]$ git cat-file -p 6d88b056751e0c2b9e008ce9eaf4a5992c559ae3
100644 blob 7c06ced7554a23812b2f7399959187a752c88add    hello.html
[ngalacan@ngalacan hello]$ git cat-file -p 7c06ced7554a23812b2f7399959187a752c88add
add
<!-- Author: N. Galatsan (1032225763@rudn.ru) -->
<html>
<head>
</head>
<body>
<h1>Hello, World!</h1>
</body>
</html>
[ngalacan@ngalacan hello]$
```

Рис. 3.23: Просмотр по хэшу

Вручную дохожу до оригинального файла `hello.html` с первого коммита по родительским хэшам и хэшу дерева (рис. 3.24).



```
ngalacan@ngalacan:~/hello
author ngalatsan <1032225763@pfur.ru> 1739461231 +0300
committer ngalatsan <1032225763@pfur.ru> 1739461289 +0300

add an author/email comment
[ngalacan@ngalacan hello]$ git cat-file -p 0d3e7b391e2bb4fefab2bc24ae748f9e5727cdb9
tree a4d5a7e4cc555b8041bcd34ad3692e96e3f266e
parent 94ba6b9c82c57085bf66ebf1491a4e5c17b1ca0a
author ngalatsan <1032225763@pfur.ru> 1739460264 +0300
committer ngalatsan <1032225763@pfur.ru> 1739460264 +0300

added HTML header
[ngalacan@ngalacan hello]$ git cat-file -p 94ba6b9c82c57085bf66ebf1491a4e5c17b1ca0a
tree 5a08fc805109b6b3add3ef11bda23005ce76f6b8
parent 2c79aeb800971f1d64fb7d8b3a29f93e287750ed
author ngalatsan <1032225763@pfur.ru> 1739460220 +0300
committer ngalatsan <1032225763@pfur.ru> 1739460220 +0300

added standart HTML page tags
[ngalacan@ngalacan hello]$ git cat-file -p 2c79aeb800971f1d64fb7d8b3a29f93e287750ed
tree b781068faad71cd6c88cb701225c3c33c6dd21bd
parent b5d59651c73af256194e619c72a567e5b825e14a
author ngalatsan <1032225763@pfur.ru> 1739460037 +0300
committer ngalatsan <1032225763@pfur.ru> 1739460037 +0300

added h1 tag
[ngalacan@ngalacan hello]$ git cat-file -p b5d59651c73af256194e619c72a567e5b825e14a
tree aaeb553f67040dbb59a4329ccde04e70b39a24b5
author ngalatsan <1032225763@pfur.ru> 1739459847 +0300
committer ngalatsan <1032225763@pfur.ru> 1739459847 +0300

Initial commit
[ngalacan@ngalacan hello]$ git cat-file -p aaeb553f67040dbb59a4329ccde04e70b39a24b5
100644 blob af5626b4a114abcb82d63db7c8082c3c4756e51b hello.html
[ngalacan@ngalacan hello]$ git cat-file -p af5626b4a114abcb82d63db7c8082c3c4756e51b
Hello, world!
[ngalacan@ngalacan hello]$
```

Рис. 3.24: Поиск первоначального файла hello.html по хэшу

## 3.15 Создание ветки

Создаю ветку style и добавляю в lib файл стилей style.css:

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

Индексирую и делаю коммит. Изменяю основную страницу, index.html, добавив поддержку стилей (рис. 3.25).

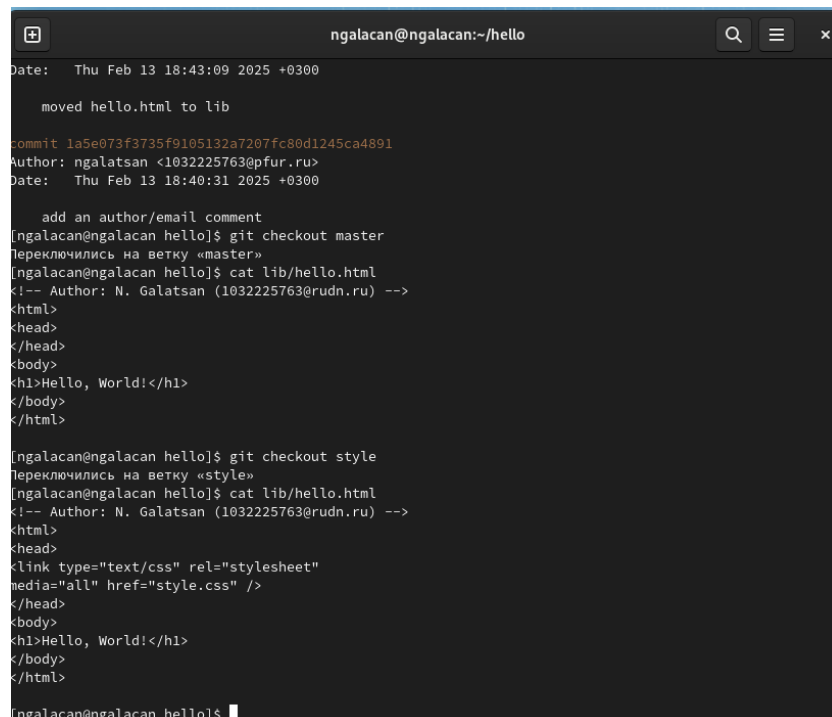
```

[ngalacan@ngalacan hello]$ git checkout -b style
Переключились на новую ветку «style»
[ngalacan@ngalacan hello]$ git status
Текущая ветка: style
ничего коммитить, нет изменений в рабочем каталоге
[ngalacan@ngalacan hello]$ touch lib/style.css
[ngalacan@ngalacan hello]$ gedit lib/style.css
[ngalacan@ngalacan hello]$ git add lib/style.css
[ngalacan@ngalacan hello]$ git commit -m 'add css stylesheet'
[style 1a29af0] add css stylesheet
1 file changed, 4 insertions(+)
create mode 100644 lib/style.css
[ngalacan@ngalacan hello]$ gedit hello.html
[ngalacan@ngalacan hello]$ gedit lib/hello.html
[ngalacan@ngalacan hello]$ git add lib/hello.html
[ngalacan@ngalacan hello]$ git commit -m 'hello uses style.css'
[style 2186f2b] hello uses style.css
1 file changed, 2 insertions(+)
[ngalacan@ngalacan hello]$ ls
index.html  lib
[ngalacan@ngalacan hello]$ gedit index.html
[ngalacan@ngalacan hello]$ git add index.html
[ngalacan@ngalacan hello]$ git commit -m 'update index.html'
[style 725e917] update index.html
1 file changed, 4 insertions(+)
[ngalacan@ngalacan hello]$

```

Рис. 3.25: Создание новой ветки и изменение файлов

В проекте теперь есть две ветки. Просматриваю лог, переключаюсь между ветками (рис. 3.26).



```

ngalacan@ngalacan: ~/hello
Date: Thu Feb 13 18:43:09 2025 +0300

    moved hello.html to lib

commit 1a5e073f3735f9105132a7207fc80d1245ca4891
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:40:31 2025 +0300

    add an author/email comment
[ngalacan@ngalacan hello]$ git checkout master
Переключились на ветку «master»
[ngalacan@ngalacan hello]$ cat lib/hello.html
<!-- Author: N. Galatsan (1032225763@rudn.ru) -->
<html>
<head>
</head>
<body>
<h1>Hello, World!</h1>
</body>
</html>

[ngalacan@ngalacan hello]$ git checkout style
Переключились на ветку «style»
[ngalacan@ngalacan hello]$ cat lib/hello.html
<!-- Author: N. Galatsan (1032225763@rudn.ru) -->
<html>
<head>
<link type="text/css" rel="stylesheet"
media="all" href="style.css" />
</head>
<body>
<h1>Hello, World!</h1>
</body>
</html>

[ngalacan@ngalacan hello]$

```

Рис. 3.26: Просмотр файлов на двух ветках

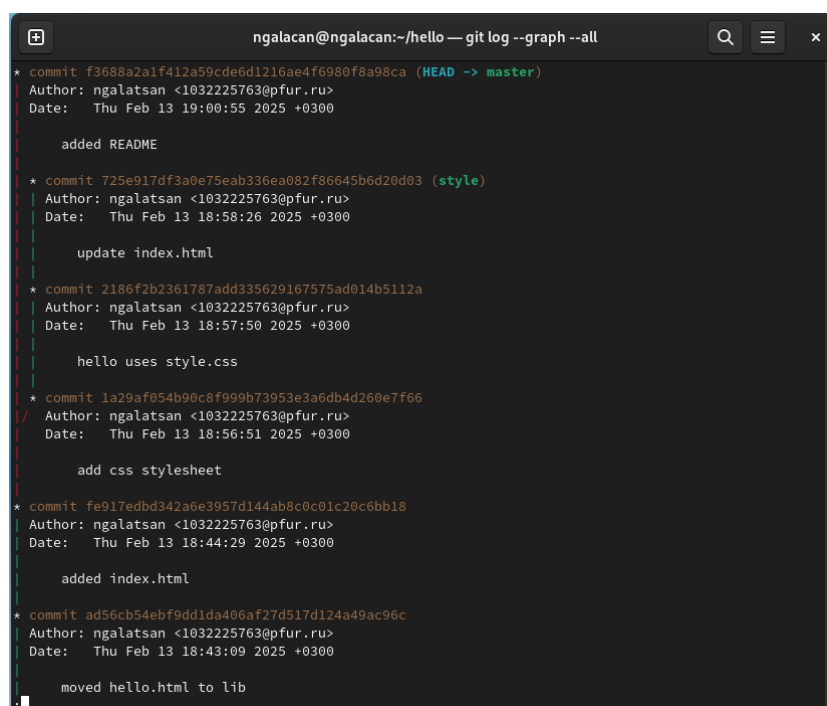
## 3.16 Изменения в ветке master

Создаю README.md в ветке master, делаю коммит изменений (рис. 3.27).

```
[ngalacan@ngalacan hello]$ git checkout master
Переключились на ветку «master»
[ngalacan@ngalacan hello]$ echo "This is the Hello World example from the git tutorial." > README.md
[ngalacan@ngalacan hello]$ git add README.md
[ngalacan@ngalacan hello]$ git commit -m 'added README'
[master f3688a2] added README
1 file changed, 1 insertion(+)
create mode 100644 README.md
[ngalacan@ngalacan hello]$
```

Рис. 3.27: Создание README.md в master

Просматриваю ветки и их отличия (рис. 3.28).



```
ngalacan@ngalacan:~/hello — git log --graph --all
* commit f3688a2a1f412a59cde6d1216ae4f6980f8a98ca (HEAD -> master)
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 19:00:55 2025 +0300

    added README

* commit 725e917df3a0e75eab336ea082f86645b6d20d03 (style)
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 18:58:26 2025 +0300

    update index.html

* commit 2186f2b2361787add335629167575ad014b5112a
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 18:57:50 2025 +0300

    hello uses style.css

* commit 1a29af054b90c8f999b73953e3a6db4d260e7f66
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 18:56:51 2025 +0300

    add css stylesheet

* commit fe917edbd342a6e3957d144ab8c0c01c20c6bb18
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 18:44:29 2025 +0300

    added index.html

* commit ad56cb54ebf9dd1da406af27d517d124a49ac96c
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 18:43:09 2025 +0300

    moved hello.html to lib
```

Рис. 3.28: Просмотр веток

## 3.17 Слияние

Слияние переносит изменения из двух веток в одну. Возвращаюсь к ветке style и сливаю master с style и просматриваю ветки (рис. 3.29).

```
ngalacan@ngalacan:~/hello — git log --graph --all
checkout      cherry      cherry-pick
[ngalacan@ngalacan hello]$ git checkout style
Переключились на ветку «style»
[ngalacan@ngalacan hello]$ git merge master
Merge made by the 'ort' strategy.
 README.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
[ngalacan@ngalacan hello]$ git log --graph --log
fatal: неопознанный аргумент: --log
[ngalacan@ngalacan hello]$ git log --graph --all
*   commit aaf2a4a8c3b24ada05bb16fc37072f6b36452271 (HEAD -> style)
|\
| | Merge: 725e917 f3688a2
| | Author: ngalatsan <1032225763@pfur.ru>
| | Date: Thu Feb 13 19:02:12 2025 +0300
| |
| | Merge branch 'master' into style
| |
| * commit f3688a2a1f412a59cde6d1216ae4f6980f8a98ca (master)
| | Author: ngalatsan <1032225763@pfur.ru>
| | Date: Thu Feb 13 19:00:55 2025 +0300
| |
| | added README
| |
| * commit 725e917df3a0e75eab336ea082f86645b6d20d03
| | Author: ngalatsan <1032225763@pfur.ru>
| | Date: Thu Feb 13 18:58:26 2025 +0300
| |
| | update index.html
| |
| * commit 2186f2b2361787add335629167575ad014b5112a
| | Author: ngalatsan <1032225763@pfur.ru>
| | Date: Thu Feb 13 18:57:50 2025 +0300
| |
| | hello uses style.css
```

Рис. 3.29: Слияние master с style

## 3.18 Создание конфликта

В ветке master изменяю hello.html:

Hello, World! Life is great!

Делаю коммит, просматриваю ветки, вижу конфликт (рис. 3.30).

```
ngalacan@ngalacan:~/hello — git log --graph --all
[ngalacan@ngalacan hello]$ git checkout master
Переключились на ветку «master»
[ngalacan@ngalacan hello]$ gedit lib/hello.html
[ngalacan@ngalacan hello]$ git add lib/hello.html
[ngalacan@ngalacan hello]$ git commit -m 'life is great'
[master bla35e5] life is great
1 file changed, 3 insertions(+), 3 deletions(-)
[ngalacan@ngalacan hello]$ git log --graph --all
* commit bla35e55575293e3933006f4e09006b693ff92f4 (HEAD -> master)
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 19:03:59 2025 +0300

    life is great

* commit aaf2a4a8c3b24ada05bb16fc37072f6b36452271 (style)
  Merge: 725e017 f3688a2
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 19:02:12 2025 +0300

    Merge branch 'master' into style

* commit f3688a2a1f412a59cde6d1216ae4f6980f8a98ca
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 19:00:55 2025 +0300

    added README
```

Рис. 3.30: Создание конфликта

Возвращаюсь к ветке `style` и пытаюсь 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`. Вношу изменения в `lib/hello.html`, оставив только необходимую запись и добавляю этот файл в репозиторий, чтобы вручную разрешить конфликт (рис. 3.31).

```
[ngalacan@ngalacan hello]$ git checkout style
Переключились на ветку «style»
[ngalacan@ngalacan hello]$ git merge master
Автослияние lib/hello.html
КОНФЛИКТ (содержимое): Конфликт слияния в lib/hello.html
Сбой автоматического слияния; исправьте конфликты, затем зафиксируйте результат.
[ngalacan@ngalacan hello]$ gedit lib/hello.html
[ngalacan@ngalacan hello]$ git add lib/hello.html
[ngalacan@ngalacan hello]$ git commit -m 'merged master fixed conflict'
[style aa28f9c] merged master fixed conflict
[ngalacan@ngalacan hello]$
```

Рис. 3.31: Разрешение конфликта и слияние

## 3.19 Сброс ветки `style`

Вернемся на ветке `style` к точке перед тем, как мы слили ее с веткой `master`. Можно сбросить ветку к любому коммиту. По сути, это изменение указателя ветки на любую точку дерева коммитов. В этом случае мы хотим вернуться в ветке `style` в точку перед слиянием с `master`. Необходимо найти последний коммит перед слиянием. Сделаю сброс по хэшу и проверю (рис. 3.32).

```

[ngalacan@ngalacan hello]$ git reset --hard 725e917df3a0e75eab336ea082f86645b6d20d03
Указатель HEAD сейчас на коммите 725e917 update index.html
[ngalacan@ngalacan hello]$ git log --graph --all
* commit b1a35e5575293e3933006f4e09006b693ff92f4 (master)
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 19:03:59 2025 +0300

    life is great

* commit f3688a2a1f412a59cde6d1216ae4f6980f8a98ca
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 19:00:55 2025 +0300

    added README

* commit 725e917df3a0e75eab336ea082f86645b6d20d03 (HEAD -> style)
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 18:58:26 2025 +0300

    update index.html

* commit 2186f2b2361787add335629167575ad014b5112a
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 18:57:50 2025 +0300

    hello uses style.css

* commit 1a29af054b90c8f999b73953e3a6db4d260e7f66
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 18:56:51 2025 +0300

```

Рис. 3.32: Сброс ветки style по хэшу

## 3.20 Сброс ветки master

Аналогичным образом сбрасываю и эту ветку (рис. 3.33).

```

[ngalacan@ngalacan hello]$ git reset --hard f3688a2a1f412a59cde6d1216ae4f6980f8a98ca
Указатель HEAD сейчас на коммите f3688a2 added README
[ngalacan@ngalacan hello]$ git log --graph --all
* commit f3688a2a1f412a59cde6d1216ae4f6980f8a98ca (HEAD -> master)
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 19:00:55 2025 +0300

    added README

* commit 725e917df3a0e75eab336ea082f86645b6d20d03 (style)
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 18:58:26 2025 +0300

    update index.html

* commit 2186f2b2361787add335629167575ad014b5112a
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 18:57:50 2025 +0300

    hello uses style.css

* commit 1a29af054b90c8f999b73953e3a6db4d260e7f66
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 18:56:51 2025 +0300

```

Рис. 3.33: Сброс ветки master по хэшу

## 3.21 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 (рис. 3.34).

```
[ngalacan@ngalacan hello]$ git checkout style
Переключились на ветку «style»
[ngalacan@ngalacan hello]$ git rebase master
Успешно перемещён и обновлён refs/heads/style.
[ngalacan@ngalacan hello]$ git log --graph
* commit 037f59046f86815cdf9cbe5f7c43fb0397a81850 (HEAD -> style)
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 18:58:26 2025 +0300

    update index.html

* commit 5bc2b86d6953a7c2de2be167ebcf068beffff73e
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 18:57:50 2025 +0300

    hello uses style.css

* commit f328346d5d8448deaa34eabc8229ac8c78559dc3
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 18:56:51 2025 +0300

    add css stylesheet

* commit f3688a2a1f412a59cde6d1216ae4f6980f8a98ca (master)
  Author: ngalatsan <1032225763@pfur.ru>
  Date: Thu Feb 13 19:00:55 2025 +0300

    added README
```

Рис. 3.34: Перебазирование

## 3.22 Слияние в ветку master

Выполняю слияние style в master. Просматриваю лог. Ветки идентичны (рис. 3.35).

```

[ngalacan@ngalacan hello]$ git checkout master
Переключились на ветку «master»
[ngalacan@ngalacan hello]$ git merge style
Обновление f3688a2..037f590
Fast-forward
 index.html      | 4 ++++
 lib/hello.html  | 2 ++
 lib/style.css   | 4 ++++
 3 files changed, 10 insertions(+)
 create mode 100644 lib/style.css
[ngalacan@ngalacan hello]$ git log
commit 037f59046f86815cdf9cbe5f7c43fb0397a81850 (HEAD -> master, style)
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:58:26 2025 +0300

    update index.html

commit 5bc2b86d6953a7c2de2be167ebcf068beffff73e
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:57:50 2025 +0300

    hello uses style.css

commit f328346d5d8448deaa34eabc8229ac8c78559dc3
Author: ngalatsan <1032225763@pfur.ru>

```

Рис. 3.35: Слияние style в master

## 3.23 Клонирование репозитория

Создаю клон первоначального репозитория. Просматриваю его. Просматриваю лог клонированного репозитория. Отображаются все файлы и коммиты оригинального репозитория (рис. 3.36).

```

[ngalacan@ngalacan hello]$ cd ..
[ngalacan@ngalacan ~]$ pwd
/home/ngalacan
[ngalacan@ngalacan ~]$ ls
bin      octave  Видео    Загрузки  Музыка    'Рабочий стол'
hello    work    Документы  Изображения  Общедоступные  Шаблоны
[ngalacan@ngalacan ~]$ git clone hello cloned_hello
Клонирование в «cloned_hello»...
готово.
[ngalacan@ngalacan ~]$ ls
bin      hello    work    Документы  Изображения  Общедоступные  Шаблоны
cloned_hello  octave  Видео    Загрузки  Музыка    'Рабочий стол'
[ngalacan@ngalacan ~]$ cd cloned_hello/
[ngalacan@ngalacan cloned_hello]$ ls
index.html  lib  README.md
[ngalacan@ngalacan cloned_hello]$ git log --all
commit 037f59046f86815cdf9cbe5f7c43fb0397a81850 (HEAD -> master, origin/style, origin/master, origin/HEAD)
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:58:26 2025 +0300

    update index.html

commit 5bc2b86d6953a7c2de2be167ebcf068beffff73e
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:57:50 2025 +0300

    hello uses style.css

commit f328346d5d8448deaa34eabc8229ac8c78559dc3
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:56:51 2025 +0300

```

Рис. 3.36: Создание клона репозитория, просмотр файлов и коммитов



## 3.24 Удаленные ветки

Просматриваю подробную информацию о ветках. Просматриваю доступные ветки, затем и удаленные. Git выводит все коммиты в оригинальный репозиторий, но ветки в удаленном репозитории не рассматриваются как локальные. Если мы хотим собственную ветку style, мы должны сами ее создать (рис. 3.37).

```
[ngalacan@ngalacan cloned_hello]$ git remote
origin
[ngalacan@ngalacan cloned_hello]$ git remote show origin
* внешний репозиторий origin
  URL для извлечения: /home/ngalacan/hello
  URL для отправки: /home/ngalacan/hello
  HEAD ветка: master
  Внешние ветки:
    master отслеживается
    style отслеживается
  Локальная ветка, настроенная для «git pull»:
    master будет слита с внешней веткой master
  Локальная ссылка, настроенная для «git push»:
    master будет отправлена в master (уже актуальна)
[ngalacan@ngalacan cloned_hello]$ git branch
* master
[ngalacan@ngalacan cloned_hello]$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/style
[ngalacan@ngalacan cloned_hello]$
```

Рис. 3.37: Просмотр веток в клоне

## 3.25 Изменение оригинального репозитория

Перейдя в оригинальный репозиторий, изменяю README.md:

This is the Hello World example from the git tutorial.

Добавляю изменение и делаю коммит.

## 3.26 Извлечение изменений

Перехожу в клон и извлекаю изменения из оригинального репозитория. На данный момент в репозитории есть все коммиты из оригинального репозитория, но они не интегрированы в локальные ветки клонированного репозитория (рис. 3.38).

```

[ngalacan@ngalacan cloned_hello]$ cd ../hello
[ngalacan@ngalacan hello]$ gedit README.md
[ngalacan@ngalacan hello]$ git add README.md
[ngalacan@ngalacan hello]$ git commit -m 'changed README in original repo'
[master 205374f] changed README in original repo
1 file changed, 1 insertion(+)
[ngalacan@ngalacan hello]$ cd ../cloned_hello
[ngalacan@ngalacan cloned_hello]$ git fetch
remote: Перечисление объектов: 5, готово.
remote: Подсчет объектов: 100% (5/5), готово.
remote: Сжатие объектов: 100% (3/3), готово.
remote: Всего 3 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
Распаковка объектов: 100% (3/3), 330 байтов | 330.00 КиБ/с, готово.
Из /home/ngalacan/hello
037f590..205374f master -> origin/master
[ngalacan@ngalacan cloned_hello]$ git log --all
commit 205374fb0d10978059686dd213f5dad2e5d103fd (origin/master, origin/HEAD)
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 19:18:16 2025 +0300

    changed README in original repo

commit 037f59046f86815cdf9cbe5f7c43fb0397a81850 (HEAD -> master, origin/style)
Author: ngalatsan <1032225763@pfur.ru>

```

Рис. 3.38: Извлечение изменений из оригинального репозитория

## 3.27 Слияние извлеченных изменений

Просматриваю README.md. Произвожу слияние и проверяю, что файл на клоне обновился. Вместо извлечения и слияния можно было использовать одну команду: `git pull` (рис. 3.39).

```

[ngalacan@ngalacan cloned_hello]$ cat README.md
This is the Hello World example from the git tutorial.
[ngalacan@ngalacan cloned_hello]$ git merge origin/master
Обновление 037f590..205374f
Fast-forward
 README.md | 1 +
1 file changed, 1 insertion(+)
[ngalacan@ngalacan cloned_hello]$ cat README.md
This is the Hello World example from the git tutorial.

[ngalacan@ngalacan cloned_hello]$ git pull
Уже актуально.
[ngalacan@ngalacan cloned_hello]$

```

Рис. 3.39: Слияние извлеченных изменений

## 3.28 Добавление ветки наблюдения

Добавляю локальную ветку, которая отслеживает удаленную ветку (рис. 3.40).

```
[ngalacan@ngalacan cloned_hello]$ git branch --track style origin/style
branch 'style' set up to track 'origin/style'.
[ngalacan@ngalacan cloned_hello]$ git branch -a
* master
  style
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/style
[ngalacan@ngalacan cloned_hello]$ git log --max-count=2
commit 205374fb0d10978059686dd213f5dad2e5d103fd (HEAD -> master, origin/master, origin/HEAD)
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 19:18:16 2025 +0300

    changed README in original repo

commit 037f59046f86815cdf0cbe5f7c43fb0397a81850 (origin/style, style)
Author: ngalatsan <1032225763@pfur.ru>
Date: Thu Feb 13 18:58:26 2025 +0300

    update index.html
[ngalacan@ngalacan cloned_hello]$
```

Рис. 3.40: Добавление ветки наблюдения

## 3.29 Чистые репозитории

Создаю чистый репозиторий. Добавляю его к оригинальному репозиторию. В файл README.md вношу изменения:

This is the Hello World example from the git tutorial.  
(Changed in the original and pushed to shared)

Добавляю изменения в master и делаю коммит. Отправляю изменения в общий репозиторий (рис. 3.41).

```
[ngalacan@ngalacan cloned_hello]$ cd ..
[ngalacan@ngalacan ~]$ git clone --bare hello hello.git
Клонирование в голый репозиторий «hello.git»...
готово.
[ngalacan@ngalacan ~]$ ls hello.git
branches  config  description  HEAD  hooks  info  objects  packed-refs  refs
[ngalacan@ngalacan ~]$ cd hello
[ngalacan@ngalacan hello]$ git remote add shared ../hello.git
[ngalacan@ngalacan hello]$ gedit README.md
[ngalacan@ngalacan hello]$ git checkout master
M README.md
Уже на «master»
[ngalacan@ngalacan hello]$ git add README.md
[ngalacan@ngalacan hello]$ git commit -m 'added shared comment to readme'
>
[master 8c8f7a2] added shared comment to readme
1 file changed, 1 insertion(+), 1 deletion(-)
[ngalacan@ngalacan hello]$ git push shared master
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 400 байтов | 400.00 КиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To ../hello.git
205374f..8c8f7a2 master -> master
[ngalacan@ngalacan hello]$
```

Рис. 3.41: Создание чистого репозитория, добавление в оригинальный, отправка изменений в общий

## 3.30 Извлечение общих изменений

Переключаюсь на клон и добавляю чистый репозиторий. С помощью `git pull shared master` извлекаю изменения. Просматриваю `README.md` (рис. 3.42).

```
[ngalacan@ngalacan hello]$ cd ../cloned_hello/
[ngalacan@ngalacan cloned_hello]$ git remote add shared ../hello.git
[ngalacan@ngalacan cloned_hello]$ git branch --track shared master
branch 'shared' set up to track 'master'.
[ngalacan@ngalacan cloned_hello]$ git pull shared master
remote: Перечисление объектов: 5, готово.
remote: Подсчет объектов: 100% (5/5), готово.
remote: Сжатие объектов: 100% (3/3), готово.
remote: Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
Распаковка объектов: 100% (3/3), 380 байтов | 380.00 КиБ/с, готово.
Из ../hello
* branch          master      -> FETCH_HEAD
* [новая ветка]    master      -> shared/master
Обновление 205374f..8c8f7a2
Fast-forward
 README.md | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
[ngalacan@ngalacan cloned_hello]$ cat README.md
This is the Hello World example from the git tutorial.
(Changed in the original and pushed to shared)
[ngalacan@ngalacan cloned_hello]$
```

Рис. 3.42: Извлечение общих изменений, проверка

## 4 Выводы

В результате выполнения работы были получены практические навыки использования системы управления версиями git.

## Список литературы

1. Git [Электронный ресурс]. Wikimedia Foundation, Inc., 2025. URL: <https://ru.wikipedia.org/wiki/Git>.