

# **Лабораторная работа №1**

**Работа с git**

Кудякова София Андреевна

# Содержание

<b>1 Цель работы</b>	<b>6</b>
<b>2 Задание</b>	<b>7</b>
<b>3 Теоретическое введение</b>	<b>8</b>
<b>4 Выполнение лабораторной работы</b>	<b>9</b>
4.1 Подготовка . . . . .	9
4.2 Создание проекта . . . . .	9
4.3 Внесение изменений . . . . .	10
4.4 История . . . . .	13
4.5 Получение старых версий . . . . .	14
4.6 Создание тегов версий . . . . .	15
4.7 Отмена локальных изменений (до индексации) . . . . .	16
4.8 Отмена проиндексированных изменений (перед коммитом) . . . . .	17
4.9 Отмена коммитов . . . . .	18
4.10 Удаление коммиттов из ветки . . . . .	19
4.11 Удаление тега oops . . . . .	20
4.12 Изменение предыдущего коммита . . . . .	21
4.13 Перемещение файлов . . . . .	22
4.14 Подробнее о структуре . . . . .	23
4.15 Git внутри: Каталог .git . . . . .	24
4.16 Работа непосредственно с объектами git . . . . .	25
4.17 Создание ветки . . . . .	27
4.18 Навигация по веткам . . . . .	28
4.19 Изменения в ветке master . . . . .	30
4.20 Слияние . . . . .	31
4.21 Создание конфликта . . . . .	31
4.22 Разрешение конфликтов . . . . .	32
4.23 Сброс ветки style . . . . .	35
4.24 Сброс ветки master . . . . .	36
4.25 Перебазирование . . . . .	38
4.26 Слияние в ветку master . . . . .	39
4.27 Клонирование репозиториев . . . . .	40
4.28 Что такое origin? . . . . .	41
4.29 Удаленные ветки . . . . .	41
4.30 Изменение оригинального репозитория . . . . .	42

4.31 Слияние извлеченных изменений . . . . .	43
4.32 Добавление ветки наблюдения . . . . .	43
4.33 Создание чистого репозитория . . . . .	43
4.34 Отправка и извлечение изменений . . . . .	44
<b>5 Выводы</b>	<b>45</b>
<b>Список литературы</b>	<b>46</b>

# Список иллюстраций

4.1 Настройка git . . . . .	9
4.2 Создание репозитория . . . . .	10
4.3 Добавление файла в репозиторий . . . . .	10
4.4 Внесение изменений в содержимое репозитория . . . . .	11
4.5 Изменение файла hello.html . . . . .	11
4.6 Изменение файла hello.html . . . . .	12
4.7 Внесение нескольких изменений в содержимое репозитория . . . . .	12
4.8 Внесение нескольких изменений в содержимое репозитория . . . . .	13
4.9 Просмотр истории . . . . .	13
4.10 Создание тегов версий . . . . .	15
4.11 Переключение по имени тега и просмотр доступных тегов . . . . .	16
4.12 Отмена локальных изменений (до индексации) . . . . .	17
4.13 Отмена проиндексированных изменений (перед коммитом) . . . . .	18
4.14 Отмена коммитов . . . . .	19
4.15 Удаление коммитов из ветки . . . . .	20
4.16 Ничего никогда не теряется . . . . .	20
4.17 Удаление тега oops . . . . .	21
4.18 Изменение файла hello.html . . . . .	21
4.19 Изменение предыдущего коммита . . . . .	22
4.20 Создание, индексация, коммит нового файла . . . . .	24
4.21 Результат открытия index.html . . . . .	24
4.22 Каталог .git . . . . .	25
4.23 Самостоятельная работа непосредственно с объектами git . . . . .	26
4.24 Создание ветки . . . . .	27
4.25 Редактирование файла hello.html . . . . .	27
4.26 Редактирование файла index.html . . . . .	28
4.27 Создание ветки . . . . .	28
4.28 Просмотр логов новой ветки . . . . .	29
4.29 Изменения в ветке master . . . . .	30
4.30 Изменения в ветке master . . . . .	30
4.31 Создание конфликта . . . . .	31
4.32 Конфликт . . . . .	33
4.33 Разрешение конфликта . . . . .	34
4.34 Коммит разрешения конфликта . . . . .	34
4.35 Поиск коммита перед слиянием . . . . .	35
4.36 Сброс ветки style . . . . .	36
4.37 Поиск коммита перед конфликтом . . . . .	37

4.38 Перебазирование . . . . .	39
4.39 Клонирование репозиториев . . . . .	40
4.40 Изменение оригинального репозитория . . . . .	42
4.41 Слияние извлеченных изменений . . . . .	43
4.42 Создание чистого репозитория . . . . .	44
4.43 Отправка изменений . . . . .	44
4.44 Извлечение изменений . . . . .	44

# **1 Цель работы**

Цель данной лабораторной работы - приобретение практических навыков работы с системой управления версиями Git.

## **2 Задание**

1. Выполнить задания из лабораторной работы
2. Оформить отчет по лабораторной работе с помощью Markdown

## **3 Теоретическое введение**

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

# 4 Выполнение лабораторной работы

Использованы материалы из [g?].

Ранее я уже работала с git, поэтому установка у меня уже осуществлена. Переходим сразу к параметрам установки окончаний строк.

## 4.1 Подготовка

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

```
sakudyakova@dk3n55 ~/work/study/2025-2026/Имитационное моделирование/simmod $ cd  
sakudyakova@dk3n55 ~ $ git config --global core.autocrlf input  
sakudyakova@dk3n55 ~ $ git config --global core.safecrlf true  
sakudyakova@dk3n55 ~ $ git config --global core.quotepath off
```

Рис. 4.1: Настройка git

## 4.2 Создание проекта

Создадим пустой каталог hello, а в нём файл с именем hello.html. В файл добавим строчку “Hello, World!”. Затем создадим git репозиторий из этого каталога, выполнив команду git init. (рис. [4.2]).

```
sakudyakova@dk3n55 ~ $ mkdir hello  
sakudyakova@dk3n55 ~ $ cd hello/  
sakudyakova@dk3n55 ~/hello $ touch hello.html  
sakudyakova@dk3n55 ~/hello $ echo "Hello, World!" > hello.html  
sakudyakova@dk3n55 ~/hello $ git init  
Инициализирован пустой репозиторий Git в /afs/.dk.sci.pfu.edu.ru/home/s/a/sakudyakova/hello/.git/
```

Рис. 4.2: Создание репозитория

Добавим файл в репозиторий и проверим статус, мы увидим сообщение о том, что коммитить нечего (рис. [4.3]).

```
sakudyakova@dk3n55 ~/hello $ git add hello.html  
sakudyakova@dk3n55 ~/hello $ git commit -m "Initial Commit"  
  
error: gpg failed to sign the data:  
[GNUPG:] KEY_CONSIDERED B009F83EAE23FCAB710CA4B99909797C7EE26FB7 2  
[GNUPG:] BEGIN_SIGNING H8  
[GNUPG:] PINENTRY_LAUNCHED 9266 gnome3 1.3.2-unknown - xterm-256color :1 - 526  
gpg: подписать не удалось: Операция отменена  
[GNUPG:] FAILURE sign 83886179  
gpg: signing failed: Операция отменена  
  
fatal: сбой записи объекта коммита  
sakudyakova@dk3n55 ~/hello $ git commit -m "Initial Commit"  
[master (корневой коммит) 5fe58b8] Initial Commit  
 1 file changed, 1 insertion(+)  
  create mode 100644 hello.html  
sakudyakova@dk3n55 ~/hello $ git status  
Текущая ветка: master  
ничего коммитить, нет изменений в рабочем каталоге
```

Рис. 4.3: Добавление файла в репозиторий

## 4.3 Внесение изменений

Изменим содержимое файла hello.html на:

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

Проверив состояние рабочего каталога увидим, что git знает, что файл hello.html был изменен, но при этом эти изменения еще не зафиксированы в репозитории. Проиндексируем изменения и снова посмотрим статус, в нем указано, что изменения пока не записаны в резервный репозиторий. И наконец закоммитим изменения, внеся их в репозиторий и снова посмотрим статус, который теперь показывает, что все изменения внесены в репозиторий (рис. [4.4]).

```

sakudyakova@dk3n55 ~/hello $ nano hello.html
sakudyakova@dk3n55 ~/hello $ git add hello.html
sakudyakova@dk3n55 ~/hello $ git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
(используйте «git restore --staged <файл>...», чтобы убрать из индекса)
изменено:      hello.html

sakudyakova@dk3n55 ~/hello $ git commit
[master f250f17] <html>          <body>          <h1>Hello, World!</h1>  </body> </html>
 1 file changed, 1 insertion(+), 1 deletion(-)
sakudyakova@dk3n55 ~/hello $ git status
Текущая ветка: master
Нечего коммитить, нет изменений в рабочем каталоге

```

Рис. 4.4: Внесение изменений в содержимое репозитория

Изменим страницу «Hello, World», чтобы она содержала стандартные теги и (рис. [4.5]).

```

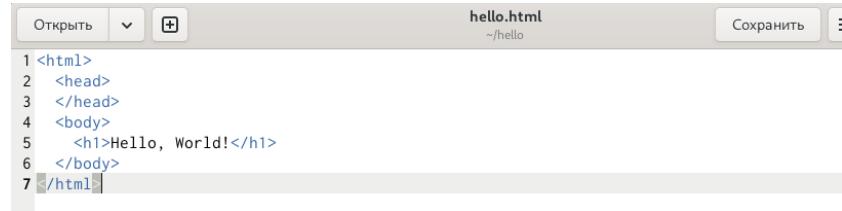
<html>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>

```



Рис. 4.5: Изменение файла hello.html

Добавим это изменение в индекс git и добавим заголовки HTML (секцию ) к странице «Hello, World». (рис. [4.6])



```
hello.html
~/hello
1 <html>
2   <head>
3   </head>
4   <body>
5     <h1>Hello, World!</h1>
6   </body>
7 </html>
```

Рис. 4.6: Изменение файла hello.html

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) не проиндексировано. (рис. [4.7]).

```
sakudyakova@dk3n55 ~/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

sakudyakova@dk3n55 ~/hello $
```

Рис. 4.7: Внесение нескольких изменений в содержимое репозитория

Произведем коммит проиндексированного изменения. Проиндексируем оставшееся изменение, посмотрим статус и прокоммитим его тоже (рис. [4.8]).

```
sakudyakova@dk3n55 ~/hello $ git commit -m"Added standard HTML page tags"
[master b7d2665] Added standard HTML page tags
 1 file changed, 5 insertions(+), 1 deletion(-)
sakudyakova@dk3n55 ~/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»)
sakudyakova@dk3n55 ~/hello $ git add .
sakudyakova@dk3n55 ~/hello $ git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    изменено:   hello.html

sakudyakova@dk3n55 ~/hello $ git commit -m"Addde"
[master a9f8eed] Added
 1 file changed, 2 insertions(+)
sakudyakova@dk3n55 ~/hello $
```

Рис. 4.8: Внесение нескольких изменений в содержимое репозитория

## 4.4 История

Получим список произведенных изменений в стандартном виде `git log`, затем в однострочном `git log --pretty=oneline` и далее с указанием времени и количества (рис. [4.10]).

```
git log --pretty=oneline --max-count=2
git log --pretty=oneline --since='5 minutes ago'
git log --pretty=oneline --until='5 minutes ago'
```

```
sakudyakova@dk3n55 ~/hello $ git log
```

```
sakudyakova@dk3n55 ~/hello $ █
```

Рис. 4.9: Просмотр истории

## 4.5 Получение старых версий

Изучим данные лога и найдем там хэш первого коммита, используя его, вернемся к первой версии репозитория и просмотрим файл hello.html, действительно, увидим первую версию. Затем вернемся к последней версии в ветке master и посмотрим на тот же файл (рис. [4.8]).

```
commit 5fe58b87d62fe19235d8ebd767bdbbf327311a0
Author: "Sofia <1132236033@pfur.ru>
Date: Thu Feb 12 17:37:54 2026 +0300

    Initial Commit
...skipping...
commit a9f8eedbbe9be94a6eadd0fc4a0a6100e6f0fcfb (HEAD -> master)
Author: "Sofia <1132236033@pfur.ru>
Date: Thu Feb 12 18:21:30 2026 +0300

    Added

commit b7d26657abeccf381ea0f7b0e91fd9470e0453b6
Author: "Sofia <1132236033@pfur.ru>
Date: Thu Feb 12 18:20:36 2026 +0300

    Added standard HTML page tags

commit f250f177df85ae90ce99985e60c0b206c7de29e3
Author: "Sofia <1132236033@pfur.ru>
Date: Thu Feb 12 17:51:45 2026 +0300

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

commit 5fe58b87d62fe19235d8ebd767bdbbf327311a0
Author: "Sofia <1132236033@pfur.ru>
Date: Thu Feb 12 17:37:54 2026 +0300

    Initial Commit
```

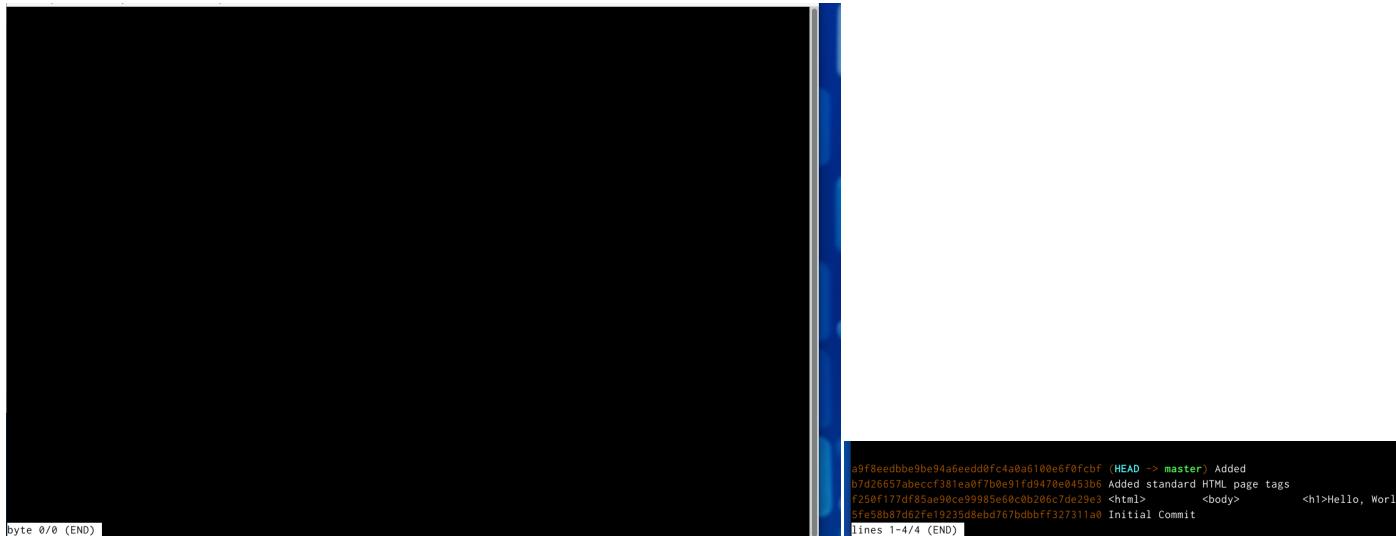
```
sakudyakova@dk3n55 ~/hello $ git log --
```

```
a9f8eedbbe9be94a6eadd0fc4a0a6100e6f0fcfb (HEAD -> master) Added
b7d26657abeccf381ea0f7b0e91fd9470e0453b6 Added standard HTML page tags
f250f177df85ae90ce99985e60c0b206c7de29e3 <html>           <body>           <h1>Hello, World!</h1>   </body> </html>
5fe58b87d62fe19235d8ebd767bdbbf327311a0 Initial Commit
lines 1-4/4 (END)
```

```
sakudyakova@dk3n55 ~/hello $ git log --pretty=
```

```
sakudyakova@dk3n55 ~/hello $
```

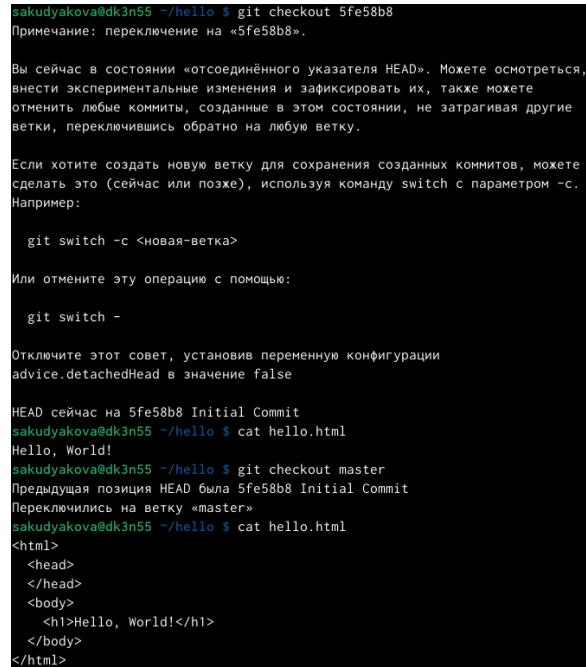
```
sakudyakova@dk3n55 ~/hello $ git log --pretty=oneline
```



```
a9f8eedbbe9be94a6eedd0fc4a0a6100e6f0fcfbf (HEAD -> master) Added
b7d26657abeccf381ea0f7b0e91fd9470e0453b6 Added standard HTML page tags
f250f177df85ae90ce99985e60c0b206c7de29e3 <html>           <body>
5fe58b87d462fe19235d8ebd767bdbbff327311a0 Initial Commit
lines 1-4/4 (END)
```

## 4.6 Создание тегов версий

Назовем текущую версию страницы hello первой (v1). Создадим тег первой версии `git tag v1` и используем его для того, чтобы вернуться к предыдущей версии, которой также присвоим тег `v1-beta` (рис. [4.10]).



```
sakudyakova@dk3n55 ~/hello $ git checkout 5fe58b8
Примечание: переключение на «5fe58b8».

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 5fe58b8 Initial Commit
sakudyakova@dk3n55 ~/hello $ cat hello.html
Hello, World!
sakudyakova@dk3n55 ~/hello $ git checkout master
Предыдущая позиция HEAD была 5fe58b8 Initial Commit
Переключились на ветку «master»
sakudyakova@dk3n55 ~/hello $ cat hello.html
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

Рис. 4.10: Создание тегов версий

Переключимся по тегам между двумя отмеченными версиями. Просмотрим все доступные теги (их два) и посмотрим теги в логе (рис. [4.11]).

```
sakudyakova@dk3n55 ~/hello $ git tag v1
sakudyakova@dk3n55 ~/hello $ git checkout v1^
git: «checkout» не является командой git. Смотрите «git --help».

Самые похожие команды:
    checkout
sakudyakova@dk3n55 ~/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 -c.
Например:

    git switch -c <новая-ветка>

Или отмените эту операцию с помощью:

    git switch -

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

HEAD сейчас на b7d2665 Added standard HTML page tags
sakudyakova@dk3n55 ~/hello $ cat hello.html
<html>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
sakudyakova@dk3n55 ~/hello $ git tag v1-beta
```

Рис. 4.11: Переключение по имени тега и просмотр доступных тегов

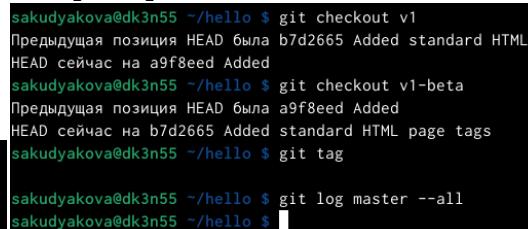
## 4.7 Отмена локальных изменений (до индексации)

Убедимся, что мы находимся на последнем коммите ветки master и внесем изменение в файл hello.html в виде нежелательного комментария (рис. [??]).



The terminal shows the current status of local changes:

```
v1
v1-beta
lines 1-2/2 (END)
```



The commit log shows the addition of the standard HTML page tags and the creation of the v1-beta tag:

```
sakudyakova@dk3n55 ~/hello $ git checkout v1
Предыдущая позиция HEAD была b7d2665 Added standard HTML
HEAD сейчас на a9f8eed Added
sakudyakova@dk3n55 ~/hello $ git checkout v1-beta
Предыдущая позиция HEAD была a9f8eed Added
HEAD сейчас на b7d2665 Added standard HTML page tags
sakudyakova@dk3n55 ~/hello $ git tag
sakudyakova@dk3n55 ~/hello $ git log master --all
sakudyakova@dk3n55 ~/hello $
```

Затем проверим статус, увидим, что изменения ещё не проиндексированы. Используем команду git checkout для переключения версии файла hello.html в

репозитории, видим, что нежелательный комментарий отсутствует в файле (рис. [4.12]).



```
1 <html>
2  <head>
3  </head>
4  <body>
5      <h1>Hello, World!</h1>
6      <!-- This is a bad comment. We want to revert it. -->
7  </body>
8 </html>
```

Рис. 4.12: Отмена локальных изменений (до индексации)

## 4.8 Отмена проиндексированных изменений (перед коммитом)

Внесем изменение в файл hello.html в виде нежелательного комментария

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

Проиндексируем это изменение и проверим состояние. Состояние показывает, что изменение было проиндексировано и готово к коммиту. Используем команду git reset, чтобы сбросить буферную зону к HEAD. Это очищает буферную зону от изменений, которые мы только что проиндексировали. Переключимся на последнюю версию коммита, посмотрев статус увидим, что наш каталог опять чист (рис. [4.13]).

```

sakudyakova@dk3n55 ~/hello $ git checkout master
error: Ваши локальные изменения в указанных файлах будут перезаписаны при переключении на состояние:
      hello.html
Сделайте коммит или спрячьте ваши изменения перед переключением веток.
Прерывая
sakudyakova@dk3n55 ~/hello $ git status
Отсоединенный указатель HEAD указывает на v1-beta
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»)
sakudyakova@dk3n55 ~/hello $ git checkout hello.html
Updated 1 path from the index
sakudyakova@dk3n55 ~/hello $ git status
Отсоединенный указатель HEAD указывает на v1-beta
ничего коммитить, нет изменений в рабочем каталоге
sakudyakova@dk3n55 ~/hello $ cat hello.html
<html>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
sakudyakova@dk3n55 ~/hello $ 

```

Рис. 4.13: Отмена проиндексированных изменений (перед коммитом)

## 4.9 Отмена коммитов

Изменим файл hello.html на следующий.

```

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

```

Проиндексируем изменения файла и прокоммитим их. Чтобы отменить коммит, нам необходимо сделать коммит, который удаляет изменения, сохраненные нежелательным коммитом. Перейдем в редактор, где изменим нежелательный коммит. Проверим лог. Проверка лога показывает нежелательные и отмененные коммиты в нашем репозитории (рис. [4.14]).

```
sakudyakova@dk3n55 ~/hello $ cat hello.html
<html>
  <head>
    <!-- This is a bad comment. We want to revert it. -->
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
sakudyakova@dk3n55 ~/hello $ git add hello.html
sakudyakova@dk3n55 ~/hello $ git status
bash: fit: команда не найдена
sakudyakova@dk3n55 ~/hello $ git status
Отсоединённый указатель HEAD указывает на v1-beta
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    изменено:   hello.html

sakudyakova@dk3n55 ~/hello $ git reset HEAD hello.html
Непройиндексированные изменения после сброса:
M     hello.html
sakudyakova@dk3n55 ~/hello $ git checkout hello.html
Updated 1 path from the index
sakudyakova@dk3n55 ~/hello $ git status
Отсоединённый указатель HEAD указывает на v1-beta
ничего коммитить, нет изменений в рабочем каталоге
sakudyakova@dk3n55 ~/hello $
```

Рис. 4.14: Отмена коммитов

## 4.10 Удаление коммиттов из ветки

Удалим последние два коммита с помощью сброса, сначала отметим последний коммит тегом, чтобы его можно было потом найти. Используем команду `git reset --hard v1`, чтобы вернуться к версии до этих коммитов. (рис. [4.15]).

```

commit 50b500a227a3d03cc199b1f70df86edff7fd000 (HEAD)
Author: "Sofia <1132236033@pfur.ru>
Date:   Wed Feb 18 15:24:48 2026 +0300

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

    This reverts commit e85cadec4edf7a645d1c37350c9543503c6df743.

commit e85cadec4edf7a645d1c37350c9543503c6df743
Author: "Sofia <1132236033@pfur.ru>
Date:   Wed Feb 18 15:24:33 2026 +0300

    Oops, we didn't want this commit

commit b7d26657abecdf381ea0f7b0e91fd9470e0453b6 (tag: v1-beta)
Author: "Sofia <1132236033@pfur.ru>
Date:   Thu Feb 12 18:20:36 2026 +0300

    Added standard HTML page tags

commit f250f177df85ae90ce99985e60c0b206c7de29e3
Author: "Sofia <1132236033@pfur.ru>
Date:   Thu Feb 12 17:51:45 2026 +0300

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

commit 5fe58b87d62fe19235d8ebd767bdbbf327311a0
Author: "Sofia <1132236033@pfur.ru>
Date:   Thu Feb 12 17:37:54 2026 +0300

    Initial Commit
lines 1-35/35 (END)

```

Рис. 4.15: Удаление коммиттов из ветки

Теперь в логе их нет, но если посмотреть логи с опцией `--all` можно их увидеть, однако метка HEAD находится на нужной нам версии (рис. [4.16]).

```

sakudyakova@dk3n55 ~/hello $ git tag oops
sakudyakova@dk3n55 ~/hello $ git reset --hard v1
Указатель HEAD сейчас на коммите a9f8eed Added
sakudyakova@dk3n55 ~/hello $ 

```

Рис. 4.16: Ничего никогда не теряется

## 4.11 Удаление тега oops

Удалим тег oops и коммиты, на которые он ссылался, сборщиком мусора. Теперь этот тег не отображается в репозитории (рис. [4.17]).

```

sakudyakova@dk3n55 - hello
commit 50b500a227a3d03cc199b1f70df86edff7fdd00d (tag: oops)
Author: "Sofia <1132236033@pfur.ru>"
Date:   Wed Feb 18 15:24:48 2026 +0300

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

    This reverts commit e85cadec4edf7a645d1c37350c9543503c6df743.

commit e85cadec4edf7a645d1c37350c9543503c6df743
Author: "Sofia <1132236033@pfur.ru>"
Date:   Wed Feb 18 15:24:33 2026 +0300

    Oops, we didn't want this commit

commit a9f8eedbbe9be94a6e0dd0fc4a0a6100e6f0fc0f (HEAD, tag: v1, master)
Author: "Sofia <1132236033@pfur.ru>"
Date:   Thu Feb 12 18:21:30 2026 +0300

    Added

commit b7d26657abecff381ea0f7b0e91fd9470e0453b6 (tag: v1-beta)
Author: "Sofia <1132236033@pfur.ru>"
Date:   Thu Feb 12 18:20:36 2026 +0300

    Added standard HTML page tags

commit f250f177df85ae90ce99985e60c0b206c7de29e3
Author: "Sofia <1132236033@pfur.ru>"
Date:   Thu Feb 12 17:51:45 2026 +0300

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

```

lines 1-36

Рис. 4.17: Удаление тега oops

## 4.12 Изменение предыдущего коммита

Добавим в страницу комментарий автора (рис. [4.18]).

```

sakudyakova@dk3n55 ~/hello $ git log --all
sakudyakova@dk3n55 ~/hello $ git tag -d oops
Метка «oops» удалена (была 50b500a)
sakudyakova@dk3n55 ~/hello $ git log --all

```

Рис. 4.18: Изменение файла hello.html

Затем добавим изменения в репозиторий. Теперь мы хотим добавить в комментарий автора почту, обновим 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. [4.19]).

Рис. 4.19: Изменение предыдущего коммита

## 4.13 Перемещение файлов

Переместим наш файл в каталог lib. Для этого создадим его и используем команду git mv, сделаем коммит этого перемещения (рис. [??]).

```
sakudyakova@dk3n55 ~/hello $ mkdir lib
sakudyakova@dk3n55 ~/hello $ git mv hello.html lib/
sakudyakova@dk3n55 ~/hello $ git status
Отсоединённый указатель HEAD на цепочке коммитов начиная с v1-beta
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    переименовано: hello.html -> lib/hello.html

sakudyakova@dk3n55 ~/hello $ git switch master
error: Ваши локальные изменения в указанных файлах будут перезаписаны при переключении на состояние:
  hello.html
Сделайте коммит или спрячьте ваши изменения перед переключением веток.
Прерывая
sakudyakova@dk3n55 ~/hello $ git status
Отсоединённый указатель HEAD на цепочке коммитов начиная с v1-beta
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    переименовано: hello.html -> lib/hello.html

sakudyakova@dk3n55 ~/hello $ git checkout master
error: Ваши локальные изменения в указанных файлах будут перезаписаны при переключении на состояние:
  hello.html
Сделайте коммит или спрячьте ваши изменения перед переключением веток.
Прерывая
sakudyakova@dk3n55 ~/hello $ git status
Отсоединённый указатель HEAD на цепочке коммитов начиная с v1-beta
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    переименовано: hello.html -> lib/hello.html

sakudyakova@dk3n55 ~/hello $ git commit -m "Moved hello.html to lib"
[отделенный HEAD 9ee4673] Moved hello.html to lib
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename hello.html => lib/hello.html (100%)
sakudyakova@dk3n55 ~/hello $
```

## 4.14 Подробнее о структуре

Добавим файл index.html в наш репозиторий, заполним файл:

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

Проиндексируем файл и сделаем коммит (рис. [4.20]).

```
sakudyakova@dk3n55 ~/hello $ touch index.html
sakudyakova@dk3n55 ~/hello $ cat index.html
cat: index: Нет такого файла или каталога
cat: html: Нет такого файла или каталога
sakudyakova@dk3n55 ~/hello $ cat index.html
sakudyakova@dk3n55 ~/hello $ cat index.html
<html>
  <body>
    <iframe src="lib/hello.html" width=200" height="200" />
  </body>
</html>
sakudyakova@dk3n55 ~/hello $ git add index.html
sakudyakova@dk3n55 ~/hello $ git commit -m "Added index.html"
[отделённый HEAD 8ac67e4] Added index.html
  1 file changed, 5 insertions(+)
create mode 100644 index.html
sakudyakova@dk3n55 ~/hello $
```

Рис. 4.20: Создание, индексация, коммит нового файла

Теперь при открытии index.html увидим кусок страницы hello в маленьком окошке (рис. [4.21]).

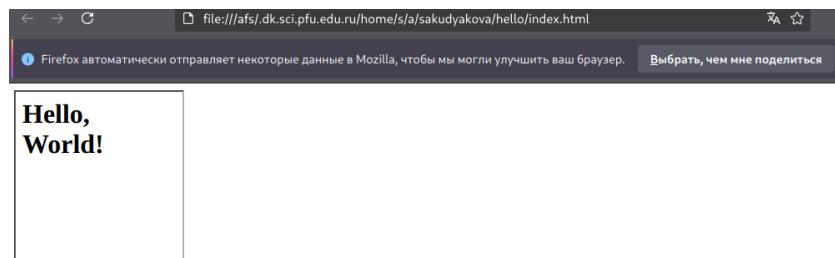


Рис. 4.21: Результат открытия index.html

## 4.15 Git внутри: Каталог .git

Просмотрим каталог, в котором хранится вся информация git. `ls -C .git/`

Затем посмотрим набор каталогов, имена которых состоят из 2 символов `ls -C .git/objects`

Имена каталогов являются первыми двумя буквами хэша sha1 объекта, хранящегося в git. Посмотрим в один из каталогов с именем из 2 букв. `ls -C .git/objects/ac` Увидим файлы с именами из 38 символов. Это файлы, содержащие объекты, хранящиеся в git.

Посмотрим файл конфигурации, создающийся для каждого конкретного

проекта cat .git/config. 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. [4.22]).

```
sakudyakova@dk3n55 ~/hello $ ls -C .git/
COMMIT_EDITMSG config description HEAD hooks index info logs objects ORIG_HEAD refs
sakudyakova@dk3n55 ~/hello $ ls -C .git/objects/
00 08 0d 28 31 3d 44 50 52 53 5f 62 63 6d 7e 8a 9e a9 ad b7 c3 e8 ec f2 f7 info pack
sakudyakova@dk3n55 ~/hello $ ls -C .git/objects/ac
ls: невозможно получить доступ к '.git/objects/ac': Нет такого файла или каталога
sakudyakova@dk3n55 ~/hello $ ls -C .git/objects/ad
2e97d369d7475f0ca497cc82aab70ed12507b6
sakudyakova@dk3n55 ~/hello $ ls -C .git/objects/ec
2bda0c449aa398508b548fe1ca8fcfa10fff
sakudyakova@dk3n55 ~/hello $ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
sakudyakova@dk3n55 ~/hello $ ls .git/refs/
heads tags
sakudyakova@dk3n55 ~/hello $ ls .git/refs/heads/
master
sakudyakova@dk3n55 ~/hello $ ls .git/refs/tags/
v1 v1-beta
sakudyakova@dk3n55 ~/hello $ ls .git/refs/tags/v1
.git/refs/tags/v1
sakudyakova@dk3n55 ~/hello $ cat .git/refs/tags/v1
a9f8beedb9e94a6eedd0fc4a0a6100ef0fcfbf
sakudyakova@dk3n55 ~/hello $ cat .git/HEAD
8ac67e4948fbbedeb71c2042c180efe392143353
sakudyakova@dk3n55 ~/hello $
```

Рис. 4.22: Каталог .git

## 4.16 Работа непосредственно с объектами git

Найдем последний коммит через log и выедем его, используя SHA1 хэш. Затем посмотрим дерево каталогов, ссылка на который идёт в последнем коммите, выведем каталог lib и файл hello.html (рис. [??]).

```

sakudyakova@dk3n55 ~%hello $ git log --max-count=1
sakudyakova@dk3n55 ~%hello $ git cat-file -t 8ac67e4948fbebeeb71c2042c180efe392143353
commit
sakudyakova@dk3n55 ~%hello $ git cat-file -p 8ac67e4948fbebeeb71c2042c180efe392143353
tree 63bf7f1fd29dd6aa793da641114c4bed139646afe
parent 9ee4673e5a5d6cfedb2ca4dbecc53cd0ca582e3
author "Sofia <1132236033@pfur.ru>" 1771418570 +0300
committer "Sofia <1132236033@pfur.ru>" 1771418570 +0300
gpgsig -----BEGIN PGP SIGNATURE-----
-----END PGP SIGNATURE-----

Added index.html
sakudyakova@dk3n55 ~%hello $ git cat-file -p 63bf7f1fb29dd6aa793da641114c4bed139646afe
100644 blob 3d78056d0ed637b699e42b57d9def6f1124e4b index.html
040000 tree ad2e97d369d7475f0ca497cc82aab706d12507b6 lib
sakudyakova@dk3n55 ~%hello $ git cat-file -p ad2e97d369d7475f0ca497cc82aab706d12507b6
100644 blob 28b42e2eb776169575ce01dfcc67fc4e5e3de715 hello.html
sakudyakova@dk3n55 ~%hello $ git cat-file -p 28b42e2eb776169579ce01dfcc67fc4e5e3de715
<!-- Author Sofia A. Kudyakova -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
sakudyakova@dk3n55 ~%hello $
```

commit 8ac67e4948fbebeeb71c2042c180efe392143353 ([HEAD](#))
Author: "Sofia <1132236033@pfur.ru>"
Date: Wed Feb 18 15:42:50 2026 +0300
Added index.html
lines 1-5/5 (END)

Исследуем git репозиторий вручную самостоятельно. Используя хэш родительского коммита последовательно дойдем до первой версии файла hello.html и посмотрим его(рис. [4.23]).

```

sakudyakova@dk3n55 ~%hello $ git log --all
sakudyakova@dk3n55 ~%hello $ git cat-file -p 5fe58b87d62fe19235d8ebd767bdbbf327311a0
tree 523cd318fc3b74ad3852cd34a74720ad6f7dd422
author "Sofia <1132236033@pfur.ru>" 1770907074 +0300
committer "Sofia <1132236033@pfur.ru>" 1770907074 +0300
gpgsig -----BEGIN PGP SIGNATURE-----
-----END PGP SIGNATURE-----

iQIZBAACAdF1EEsAn4Pq4j/KtxDKS5mQ15fH7ib7cFAMmN5c1ACgkQmQ15fH7i
b7f5gA//SeXeE28p1+ScM8HG72/6Hqxb73FD79Qn06MgK/xwvuBm+61Nq9c3gpB
nbWHvLozeYxpKsfh8WB+YGI1DinRZPBatRjYKJr7qn7H+QT61D12UtjRtv2URo84
81nJwB/VLZFTPq14xhuWuNs70Xd+GgjyxOjnMsyxBP6gIzgFV4xZDXkeZUMYc+iN
4kvKbBpwI17EZjxnjFZc//U3apR/Qv3brnmjIwcykBbd3Lhd+B=1HMozXeoI
0181zOm-TT1y8JuH7RWSohx1rATTU0use5eLnraB1fvjYeQPDq50jCFDqBQ0t
Xcw/RTmpVF4MqB/GDRLe4lpjtJUyh3vFiJ+99djaZYcl3DgwF774L4tv+D8h/nA
GRQWpC//CMZ+jbKAVoZezcPdp8LZEV7MWs+3HjZ8MxyQuRejuw21Nzz9Dzb4uj
nWDowOIje1VkjQ1Te92tdWyc6UceCxzhokl1IgjUzQeGtNGpvnHUbhKE
SR13Q0IKi54IxACKShQ0dUmDsVAtKuqXv6gsmnIBQmRxQ8ZVYnlyh1tGdz4xU
yn6ZY+h6n1qgBl4Dv19cipBQSGAdEtOr+nB3BD0JZC7ASrD8/kpBOSDad2by0FbR
Pfctm1N7SqvOzxh3XaBwRXYmB+Qqa/eb87GI1n7ETl12wKvbbxE=
=5P7M
-----END PGP SIGNATURE-----

Initial Commit
sakudyakova@dk3n55 ~%hello $ git cat-file -p 523cd318fc3b74ad3852cd34a74720ad6f7dd422
100644 blob 8ab686eafeb1f44702738c8b0f24f2567c36da6d hello.html
sakudyakova@dk3n55 ~%hello $ git cat-file -p 8ab686eafeb1f44702738c8b0f24f2567c36da6d
Hello, World!
sakudyakova@dk3n55 ~%hello $
```

Рис. 4.23: Самостоятельная работа непосредственно с объектами git

## 4.17 Создание ветки

Создадим новую ветку «style» и перейдем в неё. Добавим туда файл стилей style.css и добавим его в репозиторий (рис. [4.24]).

```
sakudyakova@dk3n55 ~/hello $ git checkout -b style
bash: git: команда не найдена
sakudyakova@dk3n55 ~/hello $ git checkout -b style
Переключились на новую ветку «style»
sakudyakova@dk3n55 ~/hello $ git status
Текущая ветка: style
ничего коммитить, нет изменений в рабочем каталоге
sakudyakova@dk3n55 ~/hello $ touch lib/style.css
sakudyakova@dk3n55 ~/hello $ cat lib/style.css
sakudyakova@dk3n55 ~/hello $ cat lib/style.css
h1 {
  color: red;
}
sakudyakova@dk3n55 ~/hello $ git add lib/style.css
sakudyakova@dk3n55 ~/hello $ git commit -m "Added css stylesheet"
[style e3c5414] Added css stylesheet
 1 file changed, 3 insertions(+)
 create mode 100644 lib/style.css
sakudyakova@dk3n55 ~/hello $
```

Рис. 4.24: Создание ветки

Обновим файл hello.html, чтобы использовать стили style.css и index.html (рис. [4.25]).

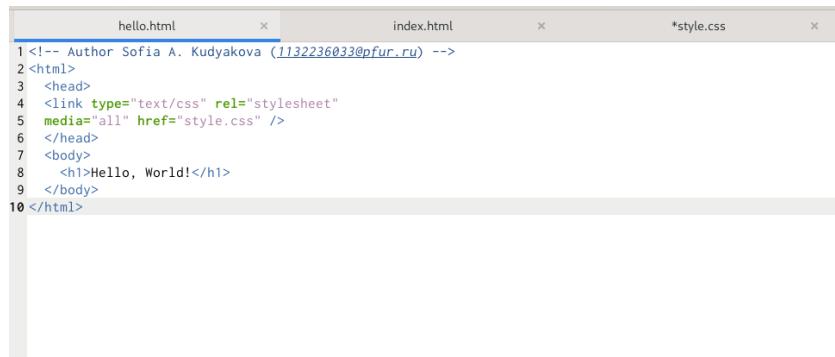
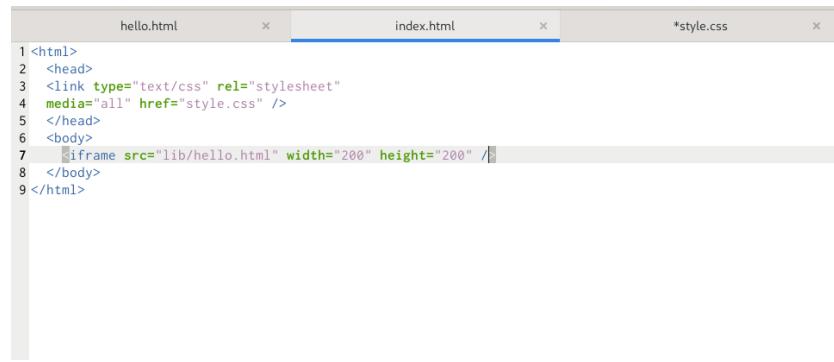


Рис. 4.25: Редактирование файла hello.html

Обновив файл index.html (рис. [4.26]).



```
1 <html>
2   <head>
3     <link type="text/css" rel="stylesheet"
4       media="all" href="style.css" />
5   </head>
6   <body>
7     <iframe src="lib/hello.html" width="200" height="200" />
8   </body>
9 </html>
```

Рис. 4.26: Редактирование файла index.html

Также обновим их в репозиторий (рис. [4.27]).

```
sakudyakova@dk3n55 ~/hello $ git add hello.html
sakudyakova@dk3n55 ~/hello $ git commit -m "Hello uses style.css"
[style 929c0ce] Hello uses style.css
 1 file changed, 10 insertions(+)
  create mode 100644 hello.html
sakudyakova@dk3n55 ~/hello $ git add index.html
sakudyakova@dk3n55 ~/hello $ git commit -m "Updated Index.html"
[style 29c9122] Updated Index.html
 1 file changed, 5 insertions(+), 1 deletion(-)
sakudyakova@dk3n55 ~/hello $
```

Рис. 4.27: Создание ветки

## 4.18 Навигация по веткам

Посмотрим все логи (рис. [4.28]).

```

commit 29c91225ad97cc12fd40deec6ac104fe903ba77a (HEAD -> style)
Author: "Sofia <1132236033@pfur.ru>
Date:   Wed Feb 18 16:20:50 2026 +0300

    Updated Index.html

commit 929c0ce1776f86120b452cb91899fbcc71a1c4a50
Author: "Sofia <1132236033@pfur.ru>
Date:   Wed Feb 18 16:19:44 2026 +0300

    Hello uses style.css

commit e3c54146250db926bf4365ed81e5c52797a5065d
Author: "Sofia <1132236033@pfur.ru>
Date:   Wed Feb 18 16:07:18 2026 +0300

    Added css stylesheet

commit 8ac67e4948fbedeeb71c2042c180efe392143353
Author: "Sofia <1132236033@pfur.ru>
Date:   Wed Feb 18 15:42:50 2026 +0300

    Added index.html

commit 9ee4673e5a5d6cfdedb2ca4dbec533cd0ca582e3
Author: "Sofia <1132236033@pfur.ru>
Date:   Wed Feb 18 15:38:24 2026 +0300

    Moved hello.html to lib

commit 4461cc0283957db84b67dfb1ff12eac2ba81ca77
Author: "Sofia <1132236033@pfur.ru>
Date:   Wed Feb 18 15:31:01 2026 +0300

    Add an author/email comment

lines 1-36

```

Рис. 4.28: Просмотр логов новой ветки

Переключимся обратно на основную ветку и просмотрим содержимое файла `lib/hello.html`, заметим, что он не использует стили, не изменился, также просмотрим содержимое этого файла в новой ветке `style`, файл в ней уже использует стили(рис. [??]).

```

sakudyakova@dk3n55 ~/hello $ git checkout master
Переключились на ветку «master»
sakudyakova@dk3n55 ~/hello $ cat lib/hello.html
cat: lib/hello.html: Нет такого файла или каталога
sakudyakova@dk3n55 ~/hello $ cat hello.html
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>

```

```

sakudyakova@dk3n55 ~/hello $ git checkout style
Переключились на ветку «style»
sakudyakova@dk3n55 ~/hello $ cat lib/hello.html
<!-- Author Sofia A. Kudyakova -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>

```

## 4.19 Изменения в ветке master

Вернемся в основную ветку и добавим файл README.md (рис. [4.29]).

```
sakudyakova@dk3n55 ~/hello $ git checkout master
Переключились на ветку «master»
sakudyakova@dk3n55 ~/hello $ echo "This is the Hello World example from the git tutorial" > README.md
sakudyakova@dk3n55 ~/hello $ git add README.md
sakudyakova@dk3n55 ~/hello $ git commit -m "Added README"
[master 5168be8] Added README
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

Рис. 4.29: Изменения в ветке master

Просмотрим ветки и их различия git log --graph --all (рис. [4.30]).

```
* commit 5168be8b9e5e39d7c93bb184938cb458a90155fc (HEAD -> master)
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 16:37:20 2026 +0300
|
|   Added README
|
* commit 8b448f1f61bf514b6a21717649acf8d4ad58881f
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 16:32:25 2026 +0300
|
|   saving
|
| * commit 29c91225ad97cc12fd40deec6ac104fe903ba77a (style)
| | Author: "Sofia <1132236033@pfur.ru>"
| | Date:   Wed Feb 18 16:20:50 2026 +0300
| |
| |   Updated Index.html
|
| * commit 929c0ce1776f86120b452cb91899fbc71a1c4a50
| | Author: "Sofia <1132236033@pfur.ru>"
| | Date:   Wed Feb 18 16:19:44 2026 +0300
| |
| |   Hello uses style.css
|
| * commit e3c54146250db926bf4365ed81e5c52797a5065d
| | Author: "Sofia <1132236033@pfur.ru>"
| | Date:   Wed Feb 18 16:07:18 2026 +0300
| |
| |   Added css stylesheet
|
| * commit 8ac67e4948fbebeeb71c2042c180efe392143353
| | Author: "Sofia <1132236033@pfur.ru>"
| | Date:   Wed Feb 18 15:42:50 2026 +0300
| |
| |   Added index.html
|
lines 1-36
```

Рис. 4.30: Изменения в ветке master

## 4.20 Слияние

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

```
* commit 5168be8b9e5e39d7c93bb184938cb458a90155fc (master)
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 16:37:20 2026 +0300
|
|     Added README
|
* commit 8b448f1f61bf514b6a21717649acf8d4ad58881f
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 16:32:25 2026 +0300
|
|     saving
|
* commit 29c91225ad97cc12fd40deec6ac104fe903ba77a (HEAD -> style)
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 16:20:50 2026 +0300
|
|     Updated Index.html
|
* commit 929c0ce1776f86120b452cb91899fbc71a1c4a50
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 16:19:44 2026 +0300
|
|     Hello uses style.css
|
* commit e3c54146250db926bf4365ed81e5c52797a5065d
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 16:07:18 2026 +0300
|
|     Added css stylesheet
|
* commit 8ac67e4948fbebedeb71c2042c180efe392143353
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 15:42:50 2026 +0300
|
|     Added index.html
|
lines 1-36
```

sakudyakova@dk3n55 ~/hello \$ git log --graph --all  
sakudyakova@dk3n55 ~/hello \$ git checkout style  
Переключились на ветку «style»  
sakudyakova@dk3n55 ~/hello \$ git merge master  
Автослияние lib/hello.html  
КОНФЛИКТ (добавление/дополнение): Конфликт слияния в lib/hello.html  
Сбой автоматического слияния; исправьте конфликты, затем зафиксируйте результат.  
sakudyakova@dk3n55 ~/hello \$ git log --graph --all

## 4.21 Создание конфликта

Вернемся в ветку master и создадим конфликт, внеся изменения в файл hello.html (рис. [4.31]).

```
1 <!-- Author Sofia A. Kudyakova (1132236033@pfur.ru) -->
2 <html>
3   <head>
4     <!-- no style -->
5 >>>>> master
6   </head>
7   <body>
8     <h1>Hello, World! Life is great!</h1>
9   </body>
10 </html> |
```

Рис. 4.31: Создание конфликта

Просмотрим ветки. После коммита «Added README» ветка master была объ-

единена с веткой style, но в настоящее время в master есть дополнительный коммит, который не был слит с style. Последнее изменение в master конфликтует с некоторыми изменениями в style (рис. [?]).

```
* commit 1f24dc2a5d1dc5a6edc727003051933c60faaf4 (style)
| Merge: 29c9122 5168be8
| Author: "Sofia <1132236033@pfur.ru>
| Date:   Wed Feb 18 16:45:14 2026 +0300
|
|     Merge branch 'master' into style
|
* commit 5168be8b9e5e39d7c93bb184938cb458a90155fc (HEAD -> master)
| Author: "Sofia <1132236033@pfur.ru>
| Date:   Wed Feb 18 16:37:20 2026 +0300
|
|     Added README
|
* commit 8b448f1f61bf514b6a21717649acf8d4ad58881f
| Author: "Sofia <1132236033@pfur.ru>
| Date:   Wed Feb 18 16:32:25 2026 +0300
|
|     saving
|
* commit 29c91225ad97cc12fd40deec6ac104fe903ba77a
| Author: "Sofia <1132236033@pfur.ru>
| Date:   Wed Feb 18 16:20:50 2026 +0300
|
|     Updated Index.html
|
* commit 929c0ce1776f86120b452cb91899fbc71a1c4a50
| Author: "Sofia <1132236033@pfur.ru>
| Date:   Wed Feb 18 16:19:44 2026 +0300
|
|     Hello uses style.css
|
* commit e3c54146250db926bf4365ed81e5c52797a5065d
| Author: "Sofia <1132236033@pfur.ru>
| Date:   Wed Feb 18 16:07:18 2026 +0300
|
|     Added css stylesheet
lines 1-36
```

```
sakudyakova@dk3n55 ~/hello $ git checkout master
Переключились на ветку «master»
sakudyakova@dk3n55 ~/hello $ git add lib/hello.html
sakudyakova@dk3n55 ~/hello $ git commit -m ''
sakudyakova@dk3n55 ~/hello $ git commit -m 'Life is great'
Текущая ветка: master
ничего коммитить, нет изменений в рабочем каталоге
sakudyakova@dk3n55 ~/hello $ git log --graph -all
```

## 4.22 Разрешение конфликтов

Вернемся к ветке style и попытаемся объединить ее с новой веткой master. В файле lib/hello.html можно увидеть записи с обеих версий этого файла. Первый раздел — версия текущей ветки (style). Второй раздел — версия ветки master. Внесем изменения в lib/hello.html, оставив только необходимую нам запись и добавим этот файл в репозиторий, чтобы вручную разрешить конфликт (рис. [4.32],[4.33], [4.34]).

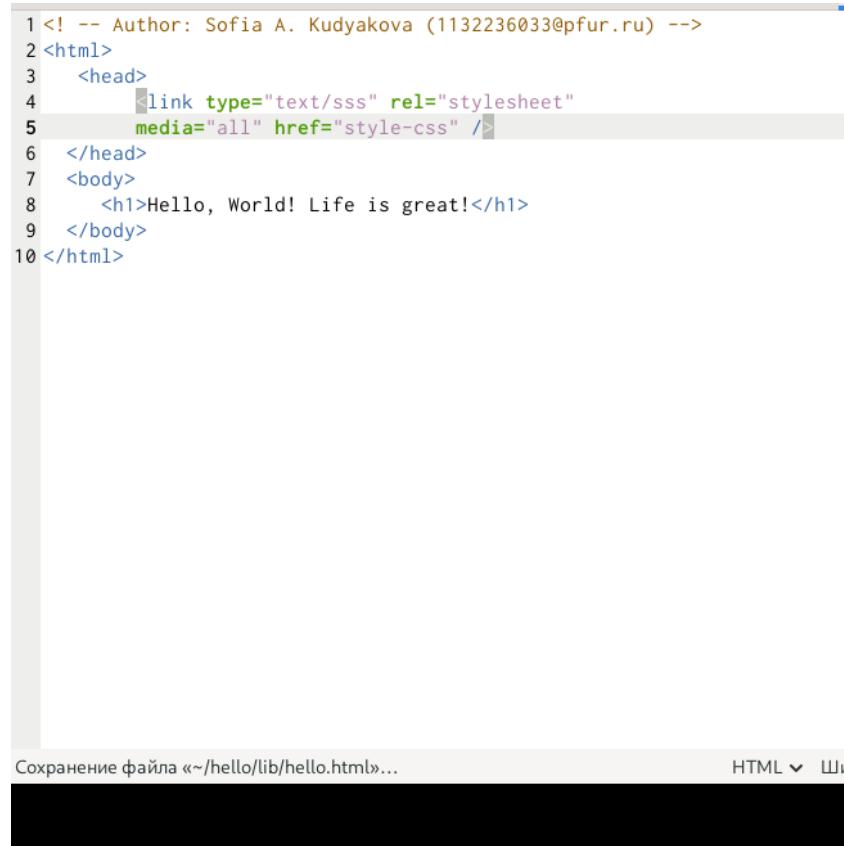
```
1 <! -- Author: Sofia A. Kudyakova (1132236033@pfur.ru) -->
2 <html>
3   <head>
4 <<<<< HEAD
5     <link type="text/css" rel="stylesheet"
6       media="all" href="style-css" />
7 =====
8     <!-- no style -->
9 >>>>> master
10   </head>
11   <body>
12     <h1>Hello, World! Life is great!</h1>
13   </body>
14 </html>
```

Bracket match found on line: 14

HTML ▾ Ш

A screenshot of a code editor interface. The main area displays an HTML document with line numbers from 1 to 14. Lines 1 through 7 are part of the head section, while lines 8 through 14 are part of the body section. Line 14 contains the closing tag for the html element. A light gray bar highlights the entire line 14. In the bottom left corner of the editor window, there is a message: "Bracket match found on line: 14". In the bottom right corner, there is a dropdown menu labeled "HTML" and some other text that is mostly obscured by a black redaction box.

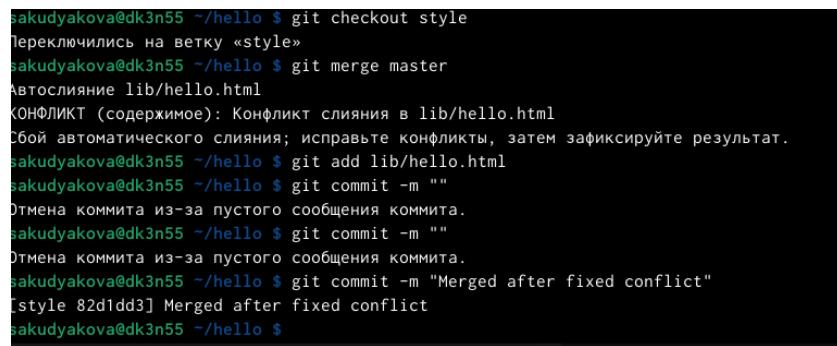
Рис. 4.32: Конфликт



The screenshot shows a code editor with a file named `lib/hello.html`. The code contains a `<link>` tag with a `media="all"` attribute. A conflict marker (`@@@`) is present in the code, indicating a merge conflict. Below the code editor is a status bar with the message "Сохранение файла «~/hello/lib/hello.html»..." and a dropdown menu labeled "HTML".

```
1 <! -- Author: Sofia A. Kudyakova (1132236033@pfur.ru) -->
2 <html>
3   <head>
4     <link type="text/css" rel="stylesheet"
5       media="all" href="style-css" />
6   </head>
7   <body>
8     <h1>Hello, World! Life is great!</h1>
9   </body>
10 </html>
```

Рис. 4.33: Разрешение конфликта



A terminal session showing the resolution of a merge conflict. The user checks out the `style` branch, merges it into `master`, and finds a conflict in `lib/hello.html`. They add the file to the index, commit the fix with an empty message, and then commit again with a message indicating the merge was after fixing the conflict.

```
sakudyakova@dk3n55 ~/hello $ git checkout style
Переключились на ветку «style»
sakudyakova@dk3n55 ~/hello $ git merge master
Автослияние lib/hello.html
КОНФЛИКТ (содержимое): Конфликт слияния в lib/hello.html
Сбой автоматического слияния; исправьте конфликты, затем зафиксируйте результат.
sakudyakova@dk3n55 ~/hello $ git add lib/hello.html
sakudyakova@dk3n55 ~/hello $ git commit -m ""
Отмена коммита из-за пустого сообщения коммита.
sakudyakova@dk3n55 ~/hello $ git commit -m ""
Отмена коммита из-за пустого сообщения коммита.
sakudyakova@dk3n55 ~/hello $ git commit -m "Merged after fixed conflict"
[style 82d1dd3] Merged after fixed conflict
sakudyakova@dk3n55 ~/hello $
```

Рис. 4.34: Коммит разрешения конфликта

## 4.23 Сброс ветки style

Вернемся на ветку style к точке перед тем, как мы слили ее с веткой master. Мы хотим вернуться в ветку style в точку перед слиянием с master. Нам необходимо найти последний коммит перед слиянием (рис. [4.35]).

```
* commit 82d1dd30578b51ddd57dae608de9c593c1bc9a3d (HEAD -> style)
| Merge: 1f24dc2 57ba7a0
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 17:02:22 2026 +0300
|
|     Merged after fixed conflict
|
* commit 57ba7a079dbe6bfce9238a9b5aecb2fe1120c281 (master)
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 17:01:13 2026 +0300
|
|     coo
|
* commit 1f24dc2a5d1dc5a6edc727003051933c60faa1f4
| Merge: 29c9122 5168be8
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 16:45:14 2026 +0300
|
|     Merge branch 'master' into style
|
* commit 5168be8b9e5e39d7c93bb184938cb458a90155fc
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 16:37:20 2026 +0300
|
|     Added README
|
* commit 8b448f1f61bf514b6a21717649acf8d4ad58881f
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 16:32:25 2026 +0300
|
|     saving
|
* commit 29c91225ad97cc12fd40deec6ac104fe903ba77a
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 16:20:50 2026 +0300
|
lines 1-36
```

Рис. 4.35: Поиск коммита перед слиянием

Мы видим, что коммит «Updated index.html» был последним на ветке style перед слиянием. Сбросим ветку style к этому коммиту.

Поищем лог ветки style. Увидим, что у нас в истории больше нет коммитов слияний (рис. [4.36]).

```
* commit 57ba079dbe6bfce9238a9b5aecb2fe1120c281 (master)
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 17:01:13 2026 +0300
|
|       coo
|
* commit 5168be8b9e5e39d7c93bb184938cb458a90155fc
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 16:37:20 2026 +0300
|
|       Added README
|
* commit 8b448f1f61bf514b6a21717649acf8d4ad58881f
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 16:32:25 2026 +0300
|
|       saving
|
| * commit 29c91225ad97cc12fd40deec6ac104fe903ba77a (HEAD -> style)
| | Author: "Sofia <1132236033@pfur.ru>"
| | Date:   Wed Feb 18 16:20:50 2026 +0300
| |
| |       Updated Index.html
|
| * commit 929c0ce1776f86120b452cb91899fbc71a1c4a50
| | Author: "Sofia <1132236033@pfur.ru>"
| | Date:   Wed Feb 18 16:19:44 2026 +0300
| |
| |       Hello uses style.css
|
| * commit e3c541462500db926bf4365ed81e5c52797a5065d
| | Author: "Sofia <1132236033@pfur.ru>"
| | Date:   Wed Feb 18 16:07:18 2026 +0300
| |
| |       Added css stylesheet
|
lines 1-36
```

Рис. 4.36: Сброс ветки style

## 4.24 Сброс ветки master

Добавив интерактивный режим в ветку master, мы внесли изменения, конфликтующие с изменениями в ветке style. Давайте вернемся в ветку master в точку перед внесением конфликтующих изменений. Это позволяет нам продемонстрировать работу команды git rebase, не беспокоясь о конфликтах. Просмотрим коммиты ветки master (рис. [4.37]).

```
* commit 57ba7a079dbe6bfce9238a9b5aecb2fe1120c281 (HEAD -> master)
| Author: "Sofia <1132236033@pfur.ru>
| Date:   Wed Feb 18 17:01:13 2026 +0300
|
|       coo
|
* commit 5168be8b9e5e39d7c93bb184938cb458a90155fc
| Author: "Sofia <1132236033@pfur.ru>
| Date:   Wed Feb 18 16:37:20 2026 +0300
|
|       Added README
|
* commit 8b448f1f61bf514b6a21717649acf8d4ad58881f
| Author: "Sofia <1132236033@pfur.ru>
| Date:   Wed Feb 18 16:32:25 2026 +0300
|
|       saving
|
* commit a9f8eedbbe9be94a6eedd0fc4a0a6100e6f0fcfb (tag: v1)
| Author: "Sofia <1132236033@pfur.ru>
| Date:   Thu Feb 12 18:21:30 2026 +0300
|
|       Added
|
* commit b7d26657abecf381ea0f7b0e91fd9470e0453b6 (tag: v1-beta)
| Author: "Sofia <1132236033@pfur.ru>
| Date:   Thu Feb 12 18:20:36 2026 +0300
|
|       Added standard HTML page tags
|
* commit f250f177df85ae90ce99985e60c0b206c7de29e3
| Author: "Sofia <1132236033@pfur.ru>
| Date:   Thu Feb 12 17:51:45 2026 +0300
|
|       <html>
|           <body>
lines 1-36
```

Рис. 4.37: Поиск коммита перед конфликтом

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

```

* commit 5168be8b9e5e39d7c93bb184938cb458a90155fc (HEAD -> master)
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 16:37:20 2026 +0300
|
|     Added README
|
* commit 8b448f1f61bf514b6a21717649acf8d4ad58881f
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 16:32:25 2026 +0300
|
|     saving
|
* commit 29c91225ad97cc12fd40deec6ac104fe903ba77a (style)
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 16:20:50 2026 +0300
|
|     Updated Index.html
|
* commit 929c0ce1776f86120b452cb91899fbc71a1c4a50
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 16:19:44 2026 +0300
|
|     Hello uses style.css
|
* commit e3c54146250db926bf4365ed81e5c52797a5065d
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 16:07:18 2026 +0300
|
|     Added css stylesheet
|
* commit 8ac67e4948bedeeb71c2042c180efe392143353
| Author: "Sofia <1132236033@pfur.ru>"
| Date:   Wed Feb 18 15:42:50 2026 +0300
|
|     Added index.html
|
lines 1-36

```

sakudyakova@dk3n55 ~/hello \$ git log --graph --all  
sakudyakova@dk3n55 ~/hello \$ git checkout master  
Переключились на ветку «master»  
sakudyakova@dk3n55 ~/hello \$ git log --graph  
sakudyakova@dk3n55 ~/hello \$ git reset --hard 5168be8b9e5e39d7c93bb184938cb458a90155fc  
Указатель HEAD сейчас на коммите 5168be8 Added README  
sakudyakova@dk3n55 ~/hello \$ git log --graph --all  
sakudyakova@dk3n55 ~/hello \$ git rebase --continue  
dropping e3c54146250db926bf4365ed81e5c52797a5065d Added css stylesheet -- patch contents already upstream  
dropping 29c91225ad97cc12fd40deec6ac104fe903ba77a Updated Index.html -- patch contents already upstream  
Успешно перемещён и обновлён refs/heads/style.  
sakudyakova@dk3n55 ~/hello \$

## 4.25 Перебазирование

Используем команду rebase вместо команды merge. Мы вернулись в точку до первого слияния и хотим перенести изменения из ветки master в нашу ветку style. На этот раз для переноса изменений из ветки master мы будем использовать команду git rebase вместо слияния (рис. [4.38]).

```
* commit 68ecbbe957c868927973706f58a53fc109e23eb7 (HEAD -> style)
| Author: "Sofia <1132236033@pfur.ru>"
| Date:  Wed Feb 18 16:19:44 2026 +0300
|
|   Hello uses style.css
|
* commit 3c24aa8c4381599456860619b679fc959752469a
| Author: "Sofia <1132236033@pfur.ru>"
| Date:  Wed Feb 18 15:31:01 2026 +0300
|
|   Add an author/email comment
|
* commit 5168be8b9e5e39d7c93bb184938cb458a90155fc (master)
| Author: "Sofia <1132236033@pfur.ru>"
| Date:  Wed Feb 18 16:37:20 2026 +0300
|
|   Added README
|
* commit 8b448f1f61bf514b6a21717649acf8d4ad58881f
| Author: "Sofia <1132236033@pfur.ru>"
| Date:  Wed Feb 18 16:32:25 2026 +0300
|
|   saving
|
* commit a9f8eedbbe9be94a6eedd0fc4a0a6100e6f0fcbf (tag: v1)
| Author: "Sofia <1132236033@pfur.ru>"
| Date:  Thu Feb 12 18:21:30 2026 +0300
|
|   Added
|
* commit b7d26657abeccf381ea0f7b0e91fd9470e0453b6 (tag: v1-beta)
| Author: "Sofia <1132236033@pfur.ru>"
| Date:  Thu Feb 12 18:20:36 2026 +0300
|
|   Added standard HTML page tags
|
lines 1-36
```

Рис. 4.38: Перебазирование

## 4.26 Слияние в ветку master

Вернемся в ветку master и сольем ветку style в неё с помощью команды git merge(рис. [??]).

```

commit 68ecbbe957c868927973706f58a53fc109e23eb7 (HEAD -> master, sty
Author: "Sofia <1132236033@pfur.ru>
Date:   Wed Feb 18 16:19:44 2026 +0300

    Hello uses style.css

commit 3c24aa8c4381599456860619b679fc959752469a
Author: "Sofia <1132236033@pfur.ru>
Date:   Wed Feb 18 15:31:01 2026 +0300

    Add an author/email comment

commit 5168be8b9e5e39d7c93bb184938cb458a90155fc
Author: "Sofia <1132236033@pfur.ru>
Date:   Wed Feb 18 16:37:20 2026 +0300

    Added README

commit 8b448f1f61bf514b6a21717649acf8d4ad58881f
Author: "Sofia <1132236033@pfur.ru>
Date:   Wed Feb 18 16:32:25 2026 +0300

    saving

commit a9f8eedbbe9be94a6eedd0fc4a0a6100e6f0fcfb (tag: v1)
Author: "Sofia <1132236033@pfur.ru>
Date:   Thu Feb 12 18:21:30 2026 +0300

    Added

commit b7d26657abecdf381ea0f7b0e91fd9470e0453b6 (tag: v1-beta)
Author: "Sofia <1132236033@pfur.ru>
Date:   Thu Feb 12 18:20:36 2026 +0300

    Added standard HTML page tags
lines 1-36

```

sakudyakova@dk3n55 ~/hello \$ git checkout master  
Переключились на ветку «master»  
sakudyakova@dk3n55 ~/hello \$ git merge style  
Обновление 5168be8..68ecbbe  
Fast-forward  
 hello.html | 10 ++++++++  
 lib/hello.html | 1 +  
 2 files changed, 11 insertions(+)  
 create mode 100644 hello.html  
sakudyakova@dk3n55 ~/hello \$

## 4.27 Клонирование репозиториев

Перейдем в наш рабочий каталог и сделаем клон репозитория hello. Просмотрев его, увидим список всех файлов на верхнем уровне оригинального репозитория README.md, index.html и lib (рис. [4.39]).

```

sakudyakova@dk3n55 ~/hello $ cd ..
sakudyakova@dk3n55 ~ $ pwd
/afs/.dk.sci.pfu.edu.ru/home/s/a/sakudyakova
sakudyakova@dk3n55 ~ $ ls
121.sh archive.tar FILES.txt input.txt '^p' tmp Документы Музыка 'Снимки'
122.sh bin GNUstep lock.file public work Загрузки Общедоступные Шаблоны
123.sh blog hello output.txt public_html Видео Изображения 'Рабочий стол'
sakudyakova@dk3n55 ~ $ git clone hello cloned_hello
Клонирование в «cloned_hello»...
готово.
sakudyakova@dk3n55 ~ $ ls
121.sh bin GNUstep output.txt tmp Загрузки 'Рабочий стол'
122.sh blog hello '^p' work Изображения 'Снимки экрана'
123.sh cloned_hello input.txt public Видео Музыка Шаблоны
archive.tar FILES.txt lock.file public_html Документы Общедоступные

```

Рис. 4.39: Клонирование репозиториев

Затем просмотрим историю репозитория и увидим список всех коммитов в новый репозиторий, и он совпадает с историей коммитов в оригинальном репозитории. Единственное отличие в названиях веток (рис. [??]).

```

commit 68ecbbe957c868927973706f58a53fc109e23eb7 (HEAD -> master, origin)
Author: "Sofia <1132236033@pfur.ru>
Date:   Wed Feb 18 16:19:44 2026 +0300

    Hello uses style.css

commit 3c24aa8c4381599456860619b679fc959752469a
Author: "Sofia <1132236033@pfur.ru>
Date:   Wed Feb 18 15:31:01 2026 +0300

    Add an author/email comment

commit 5168be8b9e5e39d7c93bb184938cb458a90155fc
Author: "Sofia <1132236033@pfur.ru>
Date:   Wed Feb 18 16:37:20 2026 +0300

    Added README

commit 8b448f1f61bf514b6a21717649acf8d4ad58881f
Author: "Sofia <1132236033@pfur.ru>
Date:   Wed Feb 18 16:32:25 2026 +0300

    saving

commit a9f8eedbbe9be94a6eedd0fc4a0a6100e6f0fcfb (tag: v1)
Author: "Sofia <1132236033@pfur.ru>
Date:   Thu Feb 12 18:21:30 2026 +0300

    Added

commit b7d26657abecff381ea0f7b0e91fd9470e0453b6 (tag: v1-beta)
Author: "Sofia <1132236033@pfur.ru>
Date:   Thu Feb 12 18:20:36 2026 +0300

    Added standard HTML page tags

sakudyakova@dk3n55 ~ $ cd cloned_hello/
sakudyakova@dk3n55 ~/cloned_hello $ ls
hello.html index.html lib README.md
sakudyakova@dk3n55 ~/cloned_hello $ git log --all

```

## 4.28 Что такое origin?

Клонированный репозиторий знает об имени по умолчанию удаленного репозитория. Посмотрим, подробную информацию об имени по умолчанию. Давайте посмотрим, можем ли мы получить более по-дробную информацию об имени по умолчанию (рис. [??]).

```

sakudyakova@dk3n55 ~/cloned_hello $ git remote
origin
sakudyakova@dk3n55 ~/cloned_hello $ git remote show origin
* внешний репозиторий origin
  URL для извлечения: /afs/dk.sci.pfu.edu.ru/home/s/a/sakudyakova/hello
  URL для отправки: /afs/dk.sci.pfu.edu.ru/home/s/a/sakudyakova/hello
  HEAD ветка: master
  Внешние ветки:
    master отслеживается
    style отслеживается
  Локальная ветка, настроенная для «git pull»:
    master будет слията с внешней веткой master
  Локальная ссылка, настроенная для «git push»:
    master будет отправлена в master (уже актуальна)

```

\* master  
lines 1-1/1 (END)

## 4.29 Удаленные ветки

Посмотрим на ветки, доступные в нашем клонированном репозитории. Можно увидеть, что в списке только ветка master. Для того, чтобы увидеть все ветки используем опцию -a (рис. [??]).

```

:sakudyakova@dk3n55 ~/cloned_hello $ git branch
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/style
lines 1-4/4 (END)

```

## 4.30 Изменение оригинального репозитория

Перейдем в репозиторий hello. Внесем изменения в файл README.md. Затем добавим их в репозиторий (рис. [4.40]).

```

sakudyakova@dk3n55 ~/cloned_hello $ cat README.md
This is the Hello World example from the git tutorial.
sakudyakova@dk3n55 ~/cloned_hello $ git add README.md
sakudyakova@dk3n55 ~/cloned_hello $ git commit -m "C
lines 1-3/3 (END)

```

Рис. 4.40: Изменение оригинального репозитория

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

```

commit 68ecbbe957c868927973706f58a53fc109e23eb7 (HEAD -> master, origin/master)
Author: "Sofia <1132236033@pfur.ru>"
Date: Wed Feb 18 16:19:44 2026 +0300

    Hello uses style.css

commit 3c24aa8c4381599456860619b679fc959752469a
Author: "Sofia <1132236033@pfur.ru>"
Date: Wed Feb 18 15:31:01 2026 +0300

    Add an author/email comment

commit 5168be8b9e5e39d7c93bb184938cb458a90155fc
Author: "Sofia <1132236033@pfur.ru>"
Date: Wed Feb 18 16:37:20 2026 +0300

    Added README

commit 8b448f1f61bf514b6a21717649acf8d4ad5881f
Author: "Sofia <1132236033@pfur.ru>"
Date: Wed Feb 18 16:32:25 2026 +0300

    saving

commit a9f8ee0bbe9be94a0eedd0fc4a0ab100e6f0fcff (tag: v1)
Author: "Sofia <1132236033@pfur.ru>"
Date: Thu Feb 12 18:21:30 2026 +0300

    Added

commit b7d26657abecdf381ea0f7b0e91fd9470e0453b6 (tag: v1-beta)
Author: "Sofia <1132236033@pfur.ru>"
Date: Thu Feb 12 18:20:36 2026 +0300

    Added standard HTML page tags
lines 1-36

```

## 4.31 Слияние извлеченных изменений

Сольем внесённые изменения в главную ветку с помощью `git pull`, которая является объединением `fetch` и `merge` в одну команду (рис. -4.41).

```
sakudyakova@dk3n55 ~/cloned_hello $ cat README.md
This is the Hello World example from the git tutorial.
sakudyakova@dk3n55 ~/cloned_hello $ git merge origin/master
Уже актуально.
sakudyakova@dk3n55 ~/cloned_hello $ cat README.md
This is the Hello World example from the git tutorial.
sakudyakova@dk3n55 ~/cloned_hello $ git pull
Уже актуально.
sakudyakova@dk3n55 ~/cloned_hello $
```

Рис. 4.41: Слияние извлеченных изменений

## 4.32 Добавление ветки наблюдения

Добавим локальную ветку, которая отслеживает удаленную ветку, теперь мы можем видеть ветку `style` в списке веток и логе (рис. [??]).

```
sakudyakova@dk3n55 ~/cloned_hello $ git merge origin/master
Уже актуально.
sakudyakova@dk3n55 ~/cloned_hello $ cat README.md
This is the Hello World example from the git tutorial.
sakudyakova@dk3n55 ~/cloned_hello $ git pull
Уже актуально.
sakudyakova@dk3n55 ~/cloned_hello $ git branch --track style origin/style
branch 'style' set up to track 'origin/style'.
sakudyakova@dk3n55 ~/cloned_hello $ git branch -a
* master
  style
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
    remotes/origin/style
lines 1-5/5 (END)
```

```
sakudyakova@dk3n55 ~/cloned_hello $ git log --max-count=2
commit 68ecbbe957c868927973706f58a53fc109e23eb7 (HEAD -> master, origin/style)
Author: "Sofia <1132236033@pfur.ru>"
Date:   Wed Feb 18 16:19:44 2026 +0300

    Hello uses style.css

commit 3c24aa8c4381599456860619b679fc959752469a
Author: "Sofia <1132236033@pfur.ru>"
Date:   Wed Feb 18 15:31:01 2026 +0300

    Add an author/email comment
lines 1-11/11 (END)
```

## 4.33 Создание чистого репозитория

Как правило, репозитории, оканчивающиеся на `.git` являются чистыми репозиториями. Создадим такой в рабочем каталоге. Затем добавим репозиторий `hello.git` к нашему оригинальному репозиторию (рис. [4.42]).

```
sakudyakova@dk3n55 ~/cloned_hello $ cd ..
sakudyakova@dk3n55 $ git clone --bare hello hello.git
Клонирование в пустой репозиторий «hello.git»...
готово.
sakudyakova@dk3n55 $ ls hello.git
config  description  HEAD  hooks  info  objects  packed-refs  refs
sakudyakova@dk3n55 $
```

Рис. 4.42: Создание чистого репозитория

## 4.34 Отправка и извлечение изменений

Так как чистые репозитории, как правило, расшариваются на каком-нибудь сетевом сервере, нам необходимо отправить наши изменения в другие репозитории. Начнем с создания изменения для отправки. Отредактируем файл README.md и сделаем коммит, затем отправим изменения в общий репозиторий. Затем извлечем изменения из общего репозитория(рис. [4.43], [4.44]).

```
sakudyakova@dk3n55 ~/hello $ cd ../cloned_hello
sakudyakova@dk3n55 ~/cloned_hello $ git remote add shared ../hello.git
sakudyakova@dk3n55 ~/cloned_hello $ git branch --track shared master
branch 'shared' set up to track 'master'.
sakudyakova@dk3n55 ~/cloned_hello $ git pull shared master
Из ../hello
 * branch      master    -> FETCH_HEAD
 * [новая ветка]  master    -> shared/master
Уже актуально.
sakudyakova@dk3n55 ~/cloned_hello $ cat README.md
This is the Hello World example from the git tutorial.
sakudyakova@dk3n55 ~/cloned_hello $
```

Рис. 4.43: Отправка изменений

```
sakudyakova@dk3n55 ~/hello $ cd ../cloned_hello
sakudyakova@dk3n55 ~/cloned_hello $ git remote add shared ../hello.git
sakudyakova@dk3n55 ~/cloned_hello $ git branch --track shared master
branch 'shared' set up to track 'master'.
sakudyakova@dk3n55 ~/cloned_hello $ git pull shared master
Из ../hello
 * branch      master    -> FETCH_HEAD
 * [новая ветка]  master    -> shared/master
Уже актуально.
sakudyakova@dk3n55 ~/cloned_hello $ cat README.md
This is the Hello World example from the git tutorial.
sakudyakova@dk3n55 ~/cloned_hello $
```

Рис. 4.44: Извлечение изменений

## **5 Выводы**

В процессе выполнения данной лабораторной работы я приобрела практические навыки работы с Git.

# **Список литературы**

::: Математическое моделирование :::