

# **Лабораторная работа №1**

**Математическое моделирование**

Александрова Ульяна Вадимовна

# Содержание

<b>1 Цель работы</b>	<b>8</b>
<b>2 Теоретическое введение</b>	<b>9</b>
<b>3 Выполнение лабораторной работы</b>	<b>11</b>
3.1 Работа с git . . . . .	11
3.1.1 1.1.1 Установка имени и электронной почты . . . . .	11
3.1.2 1.1.3 Установка отображения unicode . . . . .	11
3.1.3 1.2.1 Создайте страницу «Hello, World» . . . . .	12
3.1.4 1.2.2 Создание репозитория . . . . .	12
3.1.5 1.2.3 Добавление файла в репозиторий и проверка состояния	13
3.2 1.3 Внесение изменений . . . . .	13
3.2.1 1.3.1 «Hello, World» . . . . .	13
3.3 1.4 Индексация изменений . . . . .	14
3.3.1 1.4.1 Коммит изменений . . . . .	14
3.3.2 1.4.2 Добавление стандартных тегов страницы . . . . .	15
3.3.3 1.4.3 История . . . . .	16
3.3.4 1.4.4 Получение старых версий . . . . .	17
3.3.5 1.4.5 Создание тегов версий . . . . .	17
3.3.6 1.4.7 Просмотр тегов с помощью команды tag . . . . .	19
3.4 1.5 Отмена локальных изменений (до индексации) . . . . .	20
3.4.1 1.5.1 Переключение на ветку master и изменение hello.html	20
3.5 1.6 Отмена проиндексированных изменений (перед коммитом) . .	21
3.5.1 1.6.1 Изменение файла . . . . .	21
3.5.2 1.6.3 Сброс буферной зоны . . . . .	21
3.6 1.7 Отмена коммитов . . . . .	22
3.6.1 1.7.1 Отмена коммитов . . . . .	22
3.7 1.9 Удаление тега oops . . . . .	23
3.7.1 1.9.1 Удаление тега oops . . . . .	23
3.8 1.10 Внесение изменений в коммиты . . . . .	24
3.8.1 1.10.1 Изменение страницы . . . . .	24
3.8.2 1.10.2 Необходим email . . . . .	25
3.8.3 1.10.3 Измените предыдущий коммит . . . . .	25
3.8.4 1.10.4 Просмотр истории . . . . .	25
3.9 1.11 Перемещение файлов . . . . .	26
3.9.1 1.11.1 Перемещение в каталог lib . . . . .	26

3.10 1.13 Подробнее о структуре . . . . .	26
3.10.1 1.13.1 Добавление index.html . . . . .	26
3.10.2 1.14 Git внутри: Каталог .git . . . . .	27
3.11 1.15 Работа непосредственно с объектами git . . . . .	29
3.11.1 1.15.2 Вывод последнего коммита с помощью SHA1 хэша . . . . .	29
3.12 1.16 Создание ветки . . . . .	30
3.12.1 1.16.2 Добавьте файл стилей style.css . . . . .	31
3.12.2 1.16.3 Изменение основной страницы . . . . .	31
3.12.3 1.16.4 Изменение index.html . . . . .	32
3.13 1.17 Навигация по веткам . . . . .	32
3.13.1 1.17.1 Переключение между ветками . . . . .	33
3.13.2 1.18.1 Файл README в ветке master . . . . .	34
3.13.3 1.19.1 Просмотр отличающихся веток . . . . .	34
3.13.4 1.19.2 Текущие ветки . . . . .	34
3.14 1.20 Слияние . . . . .	35
3.14.1 1.20.1 Слияние веток . . . . .	35
3.15 1.21 Создание конфликта . . . . .	36
3.15.1 1.21.1 Вернитесь в master и создайте конфликт . . . . .	36
3.15.2 1.21.2 Просмотр веток . . . . .	37
3.16 1.22 Разрешение конфликтов . . . . .	38
3.16.1 1.22.1 Слияние master с веткой style . . . . .	38
3.16.2 1.22.2 Решение конфликта . . . . .	39
3.17 1.23 Сброс ветки style . . . . .	40
3.17.1 1.23.1 Сброс ветки style . . . . .	40
3.18 1.24 Сброс ветки master . . . . .	41
3.18.1 1.24.1 Сброс ветки master . . . . .	41
3.19 1.25 Перебазирование . . . . .	42
3.20 1.26 Слияние в ветку master . . . . .	43
3.21 1.27 Клонирование репозиториев . . . . .	44
3.21.1 1.27.1 Клонирование репозиториев . . . . .	44
3.22 1.28 Работа с клонированный репозиторием . . . . .	44
3.22.1 1.28.1 Лог . . . . .	44
3.23 1.29 Что такое origin? . . . . .	45
3.24 1.31 Изменение оригинального репозитория . . . . .	45
3.24.1 1.31.1 Внесение изменения в оригиналый репозиторий hello . . . . .	45
3.24.2 1.31.2 Извлечение изменений . . . . .	46
3.25 1.33 Добавление ветки наблюдения . . . . .	47
3.25.1 1.33.1 Добавление локальной ветки, которая отслеживает удаленную . . . . .	47
3.26 1.35 Создайте чистый репозиторий . . . . .	47
3.27 1.36 Добавление удаленного репозитория . . . . .	47
3.28 1.37 Отправка изменений . . . . .	48
3.29 1.38 Извлечение общих изменений . . . . .	48



# Список иллюстраций

3.1 Проверка работы утилиты на существующих папках . . . . .	11
3.2 Создание файла . . . . .	12
3.3 Рабочий файл . . . . .	12
3.4 Создание репозитория . . . . .	12
3.5 Добавление файла в репозиторий и проверка состояния . . . . .	13
3.6 Работа файла . . . . .	14
3.7 Индексация . . . . .	14
3.8 Коммит . . . . .	15
3.9 Изменение файла . . . . .	15
3.10 Изменение файла . . . . .	15
3.11 Статус . . . . .	16
3.12 Коммит . . . . .	16
3.13 Лог . . . . .	17
3.14 Возвращение к предыдущей версии . . . . .	17
3.15 Переход между версиями . . . . .	17
3.16 Тег ветки . . . . .	18
3.17 Переключение между версиями . . . . .	19
3.18 Просмотр тегов . . . . .	20
3.19 Переключение . . . . .	20
3.20 Сброс буферной зоны . . . . .	22
3.21 Отмена коммита . . . . .	23
3.22 Полный лог . . . . .	23
3.23 Удаление тега . . . . .	24
3.24 Добавление авторства . . . . .	24
3.25 Добавление почты . . . . .	25
3.26 Изменение предыдущего коммита . . . . .	25
3.27 Лог после изменения коммита . . . . .	26
3.28 Каталог lib . . . . .	26
3.29 index.html . . . . .	27
3.30 Внутренний каталог . . . . .	28
3.31 сае . . . . .	29
3.32 работа с хэш-ключами . . . . .	30
3.33 переключение . . . . .	30
3.34 Изменение файла . . . . .	31
3.35 Измененная страничка . . . . .	31
3.36 Коммиты . . . . .	32
3.37 Каталог . . . . .	33

3.38 Переключение между ветками . . . . .	34
3.39 Создание README . . . . .	34
3.40 Новая ветка . . . . .	35
3.41 Слияние веток . . . . .	36
3.42 Слияние веток лог . . . . .	36
3.43 Изменение файла . . . . .	37
3.44 Конфликт веток . . . . .	38
3.45 Файл . . . . .	39
3.46 Изменение файла . . . . .	39
3.47 Сброс ветки . . . . .	40
3.48 Ветка master . . . . .	41
3.49 Сброс ветки master . . . . .	42
3.50 Перебазирование . . . . .	43
3.51 Слияние в ветку master . . . . .	43
3.52 Лог после клонирования . . . . .	44
3.53 Слияние в ветку master . . . . .	45
3.54 Извлечение изменений . . . . .	46
3.55 Изменения . . . . .	46
3.56 Изменения . . . . .	47
3.57 Изменения . . . . .	48

# **Список таблиц**

# **1 Цель работы**

Целью данное работы является изучение и освоение утилиты Git.

## 2 Теоретическое введение

Git – это распределённая система управления версиями, которая позволяет отслеживать изменения в файлах и координировать работу нескольких людей над проектом. Она была разработана Линусом Торвальдсом в 2005 году для управления разработкой ядра Linux.

Основные особенности Git:

1. **Распределённость:** Каждый разработчик имеет полную копию репозитория, включая всю историю изменений. Это позволяет работать автономно и синхронизировать изменения позже.
2. **Быстрота:** Git оптимизирован для высокой скорости выполнения операций.
3. **Эффективность хранения:** Git использует дельты и сжатие для эффективного хранения данных.
4. **Поддержка ветвления и слияния:** Git позволяет легко создавать и объединять ветки, что упрощает работу над новыми функциями и исправлением ошибок.

Основные команды Git:

- `git init`: Инициализация нового репозитория.
- `git clone`: Клонирование существующего репозитория.
- `git add`: Добавление изменений в индекс (стейджинг).
- `git commit`: Фиксация изменений в репозитории.
- `git push`: Отправка изменений в удалённый репозиторий.

- `git pull`: Получение изменений из удалённого репозитория и их слияние с локальной копией.

# 3 Выполнение лабораторной работы

## 3.1 Работа с git

### 3.1.1 1.1.1 Установка имени и электронной почты

У меня уже был подключен гитхаб на этой машине при помощи следующих команд:

```
git config --global user.name "Your Name"  
git config --global user.email "your_email@whatever.com"
```

Как мы можем видеть, git работает успешно (рис. 3.1).

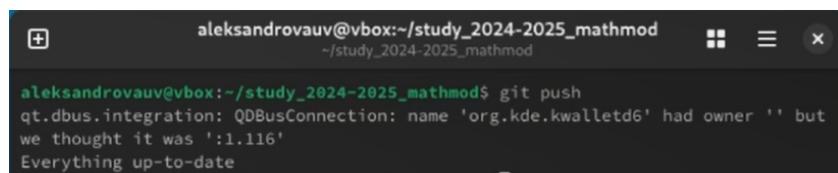
A screenshot of a terminal window titled 'aleksandrova@vbox:~/study\_2024-2025\_mathmod'. The window shows the command 'git push' being run, followed by an error message about a QDBusConnection issue, and finally the message 'Everything up-to-date'.

Рис. 3.1: Проверка работы утилиты на существующих папках

### 3.1.2 1.1.3 Установка отображения unicode

По умолчанию, git будет печатать не-ASCII символы в именах файлов в виде восьмеричных последовательностей \ппп. Чтобы избежать нечитаемых строк, я установила соответствующий флаг.

```
git config --global core.quotePath off
```

### 3.1.3 1.2.1 Создайте страницу «Hello, World»

Начну работу в пустом рабочем каталоге с создания пустого каталога с именем hello, затем войду в него и создам там файл с именем hello.html (рис. 3.2) (рис. 3.3).

```
echo "Hello, World!" > hello.html
```

```
aleksandrovauv@vbox:~/study_2024-2025_mathmod$ cd ~
aleksandrovauv@vbox:~$ mkdir hello
aleksandrovauv@vbox:~$ cd hello
aleksandrovauv@vbox:~/hello$ touch hello.html
aleksandrovauv@vbox:~/hello$ echo "Hello, World" > hello.html
```

Рис. 3.2: Создание файла

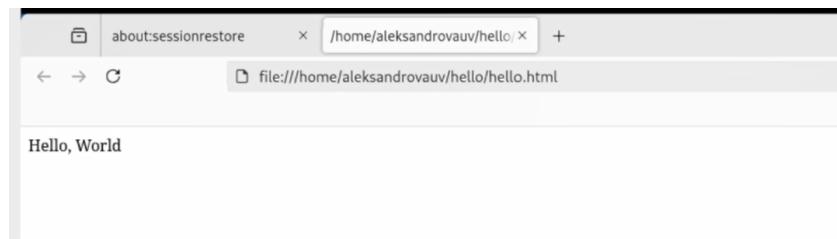


Рис. 3.3: Рабочий файл

### 3.1.4 1.2.2 Создание репозитория

Чтобы создать git репозиторий из этого каталога, выполняю команду git init (рис. 3.4).

```
aleksandrovauv@vbox:~/hello$ git init
hint: Using 'master' as the name for the initial branch. This default branch nam
e
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/aleksandrovauv/hello/.git/
```

Рис. 3.4: Создание репозитория

### **3.1.5 1.2.3 Добавление файла в репозиторий и проверка состояния**

Добавляю файл в репозиторий.

Использую команду `git status`, проверяю текущее состояние репозитория.

```
git status
```

Коммитить нечего. Это означает, что в репозитории хранится текущее состояние рабочего каталога, и нет никаких изменений, ожидающих записи (рис. 3.5).

```
aleksandrovauv@vbox:~/hello$ git add hello.html
aleksandrovauv@vbox:~/hello$ git commit -m "Initial commit"
[master (root-commit) 7e6261b] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 hello.html
aleksandrovauv@vbox:~/hello$ git status
On branch master
nothing to commit, working tree clean
aleksandrovauv@vbox:~/hello$ nano hello.html
```

Рис. 3.5: Добавление файла в репозиторий и проверка состояния

Вот текст страниц 3-5 вашего PDF-документа в формате Markdown:

## **3.2 1.3 Внесение изменений**

### **3.2.1 1.3.1 «Hello, World»**

Добавляю HTML-теги к нашему приветствию. Изменяю содержимое файла `hello.html` на (рис. 3.6).

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

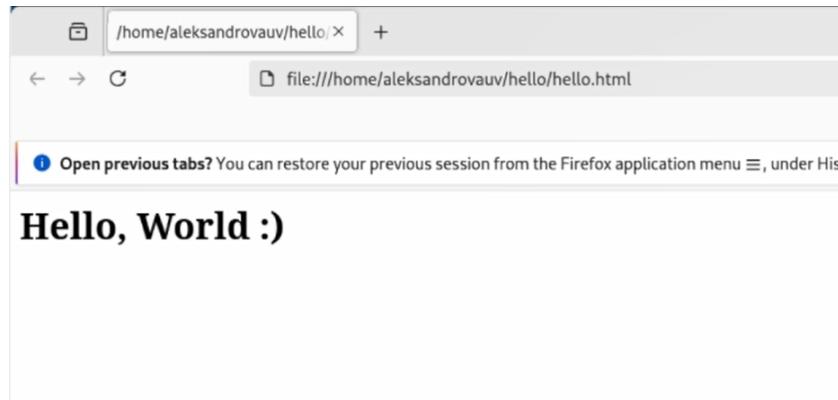


Рис. 3.6: Работа файла

Проверяю состояние рабочего каталога.

### 3.3 1.4 Индексация изменений

Теперь выполняю команду `git add`, чтобы проиндексировать изменения (рис. 3.7).

```
aleksandroauv@vbox:~/hello$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   hello.html

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

Рис. 3.7: Индексация

Изменения файла `hello.html` были проиндексированы. Это означает, что Git теперь знает об изменении, но изменение пока не записано в репозиторий. Следующий коммит будет включать в себя проиндексированные изменения.

#### 3.3.1 1.4.1 Коммит изменений

Делаю коммит и проверяю состояние (рис. 3.8).

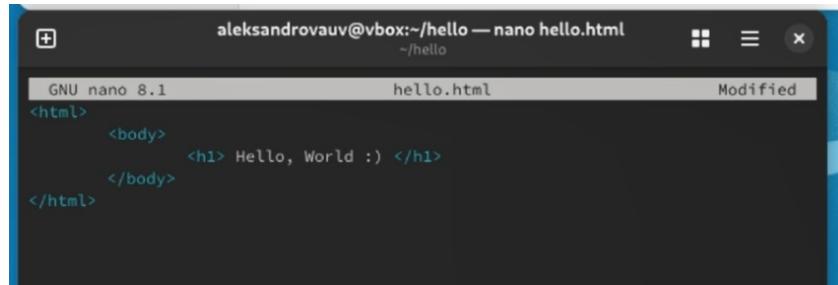
```
aleksandrova@vbox:~/hello$ git commit  
[master 0806878] Added h1 tag  
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Рис. 3.8: Коммит

Рабочий каталог чистый, можно продолжить работу.

### 3.3.2 1.4.2 Добавление стандартных тегов страницы

Изменяю страницу «Hello, World», чтобы она содержала стандартные теги <html> и <body> (рис. 3.9).



```
GNU nano 8.1          aleksandrova@vbox:~/hello — nano hello.html  
~/hello  
Modified  
<html>  
  <body>  
    <h1> Hello, World :) </h1>  
  </body>  
</html>
```

Рис. 3.9: Изменение файла

Теперь добавлю это изменение в индекс Git.

```
git add hello.html
```

Теперь добавляю заголовки HTML к странице «Hello, World» (рис. 3.10).



```
GNU nano 8.1          aleksandrova@vbox:~/hello — nano hello.html  
~/hello  
Modified  
<html>  
  <head>  
  </head>  
  <body>  
    <h1> Hello, World :) </h1>  
  </body>  
</html>
```

Рис. 3.10: Изменение файла

Проверяю текущий статус (рис. 3.11).

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

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   hello.html
```

Рис. 3.11: Статус

hello.html указан дважды в состоянии. Первое изменение (добавление стандартных тегов) проиндексировано и готово к коммиту. Второе изменение (добавление заголовков HTML) является непроиндексированным. Если бы я сделала коммит сейчас, заголовки не были бы сохранены в репозиторий.

Коммичу проиндексированное изменение, а затем еще раз проверяю состояние (рис. 3.12).

```
aleksandrovauv@vbox:~/hello$ git commit -m "Added standard HTML page tags"
[master 2bebdd6] Added standard HTML page tags
 1 file changed, 5 insertions(+), 1 deletion(-)
aleksandrovauv@vbox:~/hello$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   hello.html

no changes added to commit (use "git add" and/or "git commit -a")
```

Рис. 3.12: Коммит

Состояние команды говорит о том, что hello.html имеет незафиксированные изменения, но уже не в буферной зоне.

### 3.3.3 1.4.3 История

Получим список произведенных изменений (рис. 3.13).

```
aleksandrova@vbox:~/hello$ git log --pretty=oneline
c05d23f9940fec5660d3ea0426b5a7163cf43a21 (HEAD -> master) Adde HTML header
2bebdd6c8d9329911eb40e329d66cb6c7b01d603 Added standard HTML page tags
080687816e83796ce4775a79faef558b42f4eb1 Added h1 tag
7e6261b5504f36fb103b9d6d10c6793c64bddice Initial commit
aleksandrova@vbox:~/hello$ git log --pretty=oneline --max-count=2
c05d23f9940fec5660d3ea0426b5a7163cf43a21 (HEAD -> master) Adde HTML header
2bebdd6c8d9329911eb40e329d66cb6c7b01d603 Added standard HTML page tags
```

Рис. 3.13: Лог

### 3.3.4 1.4.4 Получение старых версий

Возвращаться назад в историю очень просто. Команда `checkout` скопирует любой снимок из репозитория в рабочий каталог (рис. 3.14).

```
aleksandrova@vbox:~/hello$ git checkout 7e6261b5504f36fb10
Note: switching to '7e6261b5504f36fb10'.
```

Рис. 3.14: Возвращение к предыдущей версии

Просматриваю файл `hello.html` той версии и возвращаюсь к последней версии в ветке `master` (рис. 3.15).

```
aleksandrova@vbox:~/hello$ cat hello.html
Hello, World
aleksandrova@vbox:~/hello$ git checkout master
Previous HEAD position was 7e6261b Initial commit
Switched to branch 'master'
aleksandrova@vbox:~/hello$ cat hello.html
<html>
  <head>
  </head>
  <body>
    <h1> Hello, World :) </h1>
  </body>
</html>
```

Рис. 3.15: Переход между версиями

### 3.3.5 1.4.5 Создание тегов версий

Называю текущую версию страницы `hello` первой (`v1`) (рис. 3.16).

```
aleksandrovauv@vbox:~/hello$ git tag v1
aleksandrovauv@vbox:~/hello$ git checkout v1^
Note: switching to 'v1^'.
```

Рис. 3.16: Тег ветки

Теперь текущая версия страницы называется v1.

### 3.3.5.1 Теги для предыдущих версий и переключение между версиями

Создаю тег для предыдущей версии через команду:

```
git checkout v1^
cat hello.html
```

Это версия с тегами <html> и <body>, но еще пока без <head>. Делаю ее версией v1-beta.

```
git tag v1-beta
```

Теперь пробую попереключаться между двумя отмеченными версиями (рис. 3.17).

```
aleksandroauv@vbox:~/hello$ cat hello.html
<html>
  <body>
    <h1> Hello, World :) </h1>
  </body>
</html>
aleksandroauv@vbox:~/hello$ git tag v1-beta
aleksandroauv@vbox:~/hello$ git checkout v1
Previous HEAD position was 2bebdd6 Added standard HTML page tags
HEAD is now at c05d23f Adde HTML header
aleksandroauv@vbox:~/hello$ cat hello.html
<html>
  <head>
  </head>
  <body>
    <h1> Hello, World :) </h1>
  </body>
</html>
aleksandroauv@vbox:~/hello$ git checkout v1-beta
Previous HEAD position was c05d23f Adde HTML header
HEAD is now at 2bebdd6 Added standard HTML page tags
aleksandroauv@vbox:~/hello$ cat hello.html
<html>
  <body>
    <h1> Hello, World :) </h1>
  </body>
</html>
```

Рис. 3.17: Переключение между версиями

### 3.3.6 1.4.7 Просмотр тегов с помощью команды tag

Просматриваю теги (рис. 3.18).

```
aleksandrova@vbox:~/hello$ git tag
v1
v1-beta
aleksandrova@vbox:~/hello$ git log master --all
commit c05d23f9940fec5660d3ea0426b5a7163cf43a21 (HEAD, tag: v1, master)
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Thu Feb 13 20:19:50 2025 +0300

    Adde HTML header

commit 2bebdd6c8d9329911eb40e329d66cb6c7b01d603 (tag: v1-beta)
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Thu Feb 13 20:18:35 2025 +0300

    Added standard HTML page tags

commit 080687816e83796ce4775a79faefd558b42f4eb1
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Thu Feb 13 20:14:11 2025 +0300

    Added h1 tag

commit 7e6261b5504f36fb103b9d6d10c6793c64bdd1ce
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Thu Feb 13 20:09:34 2025 +0300

Initial commit
```

Рис. 3.18: Просмотр тегов

Мы можем видеть теги (v1 и v1-beta) в логе вместе с именем ветки (master). Кроме того, HEAD показывает коммит, на который мы переключились (на данный момент это v1-beta).

## 3.4 1.5 Отмена локальных изменений (до индексации)

### 3.4.1 1.5.1 Переключение на ветку master и изменение hello.html

Иногда случается, что мы изменили файл в рабочем каталоге, и хотим отменить последние коммиты (рис. 3.19).

```
aleksandrova@vbox:~/hello$ git checkout master
Already on 'master'
aleksandrova@vbox:~/hello$ nano hello.html
```

Рис. 3.19: Переключение

Вношу изменение в файл hello.html в виде нежелательного комментария.

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

## 3.5 1.6 Отмена проиндексированных изменений (перед коммитом)

### 3.5.1 1.6.1 Изменение файла

Вношу 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>
```

Проиндексирую это изменение.

```
git add hello.html
```

### 3.5.2 1.6.3 Сброс буферной зоны

Команда `git reset` сбрасывает буферную зону к HEAD. Это очищает буферную зону от изменений, которые мы только что проиндексировали. Команда `git`

`reset` (по умолчанию) не изменяет рабочий каталог. Поэтому рабочий каталог все еще содержит нежелательный комментарий (рис. 3.20).

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

aleksandrova@vbox:~/hello$ git reset HEAD hello.html
Unstaged changes after reset:
M       hello.html
aleksandrova@vbox:~/hello$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   hello.html

no changes added to commit (use "git add" and/or "git commit -a")
aleksandrova@vbox:~/hello$ git checkout hello.html
Updated 1 path from the index
aleksandrova@vbox:~/hello$ git status
On branch master
nothing to commit, working tree clean
```

Рис. 3.20: Сброс буферной зоны

## 3.6 1.7 Отмена коммитов

### 3.6.1 1.7.1 Отмена коммитов

Изменяю файл `hello.html` на следующий.

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

Произвожу коммит и отменяю через `git revert HEAD` и просматриваю лог (рис. 3.21) (рис. 3.22).

```
aleksandrovauv@vbox:~/hello$ nano hello.html
aleksandrovauv@vbox:~/hello$ git add hello.html
aleksandrovauv@vbox:~/hello$ git commit -m "Oops, we didn't want to commit this"
[master 9f00528] Oops, we didn't want to commit this
 1 file changed, 1 insertion(+)
aleksandrovauv@vbox:~/hello$ git revert HEAD
[master 67264f0] Revert "Oops, we didn't want to commit this"
 1 file changed, 1 deletion(-)
aleksandrovauv@vbox:~/hello$ git log
commit 67264f026502f159c3ff81133bba6d4b969f3ee3 (HEAD -> master)
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Thu Feb 13 20:39:12 2025 +0300
      |
      Revert "Oops, we didn't want to commit this"

      This reverts commit 9f005280ed8ac02974c342975ce1d82d92fa39eb.

commit 9f005280ed8ac02974c342975ce1d82d92fa39eb
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Thu Feb 13 20:38:38 2025 +0300
```

Рис. 3.21: Отмена коммита

```
commit 67264f026502f159c3ff81133bba6d4b969f3ee3 (tag: oops)
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Thu Feb 13 20:39:12 2025 +0300

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

      This reverts commit 9f005280ed8ac02974c342975ce1d82d92fa39eb.

commit 9f005280ed8ac02974c342975ce1d82d92fa39eb
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Thu Feb 13 20:38:38 2025 +0300

      Oops, we didn't want to commit this

commit c05d23f9940fec5660d3ea0426b5a7163cf43a21 (HEAD -> master, tag: v1)
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Thu Feb 13 20:19:50 2025 +0300

      Adde HTML header

commit 2bebdd6c8d9329911eb40e329d66cb6c7b01d603 (tag: v1-beta)
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Thu Feb 13 20:18:35 2025 +0300
:
```

Рис. 3.22: Полный лог

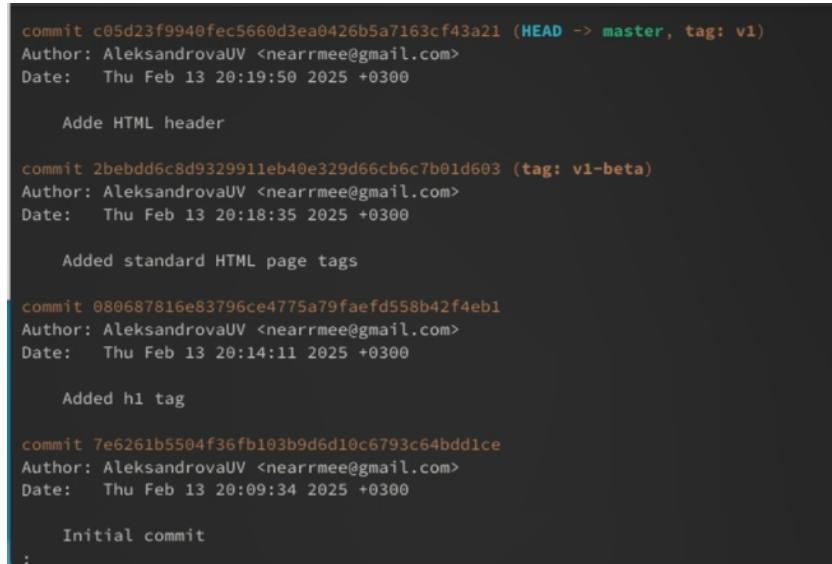
## 3.7 1.9 Удаление тега oops

### 3.7.1 1.9.1 Удаление тега oops

Тег oops свою функцию выполнил. Удалю его и коммиты, на которые он ссылался, сборщиком мусора.

```
git tag -d oops  
git log --all
```

Тег «oops» больше не будет отображаться в репозитории (рис. 3.23).



```
commit c05d23f9940fec5660d3ea0426b5a7163cf43a21 (HEAD -> master, tag: v1)
Author: AleksandrovaUV <nearrmee@gmail.com>
Date: Thu Feb 13 20:19:50 2025 +0300

    Adde HTML header

commit 2bebdd6c8d9329911eb40e329d66cb6c7b01d603 (tag: v1-beta)
Author: AleksandrovaUV <nearrmee@gmail.com>
Date: Thu Feb 13 20:18:35 2025 +0300

    Added standard HTML page tags

commit 080687816e83796ce4775a79faefd558b42f4eb1
Author: AleksandrovaUV <nearrmee@gmail.com>
Date: Thu Feb 13 20:14:11 2025 +0300

    Added h1 tag

commit 7e6261b5504f36fb103b9d6d10c6793c64bdd1ce
Author: AleksandrovaUV <nearrmee@gmail.com>
Date: Thu Feb 13 20:09:34 2025 +0300

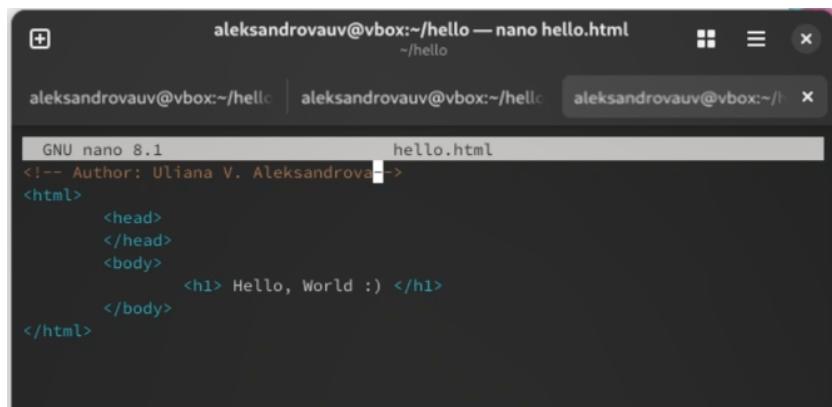
    Initial commit
:
```

Рис. 3.23: Удаление тега

## 3.8 1.10 Внесение изменений в коммиты

### 3.8.1 1.10.1 Изменение страницы

Добавляю в страницу комментарий автора (рис. 3.24).



```
aleksandrovauv@vbox:~/hello — nano hello.html
~/hello

aleksandrovauv@vbox:~/hello | aleksandrovauv@vbox:~/hello | aleksandrovauv@vbox:~/| x

GNU nano 8.1          hello.html

<html>
  <head>
  </head>
  <body>
    <h1> Hello, World :) </h1>
  </body>
</html>
```

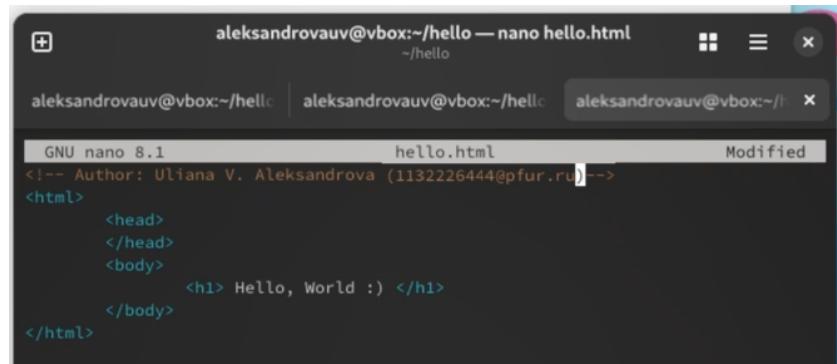
Рис. 3.24: Добавление авторства

Выполняю:

```
git add hello.html  
git commit -m "Add an author comment"
```

### 3.8.2 1.10.2 Необходим email

Добавляю email (рис. 3.25).

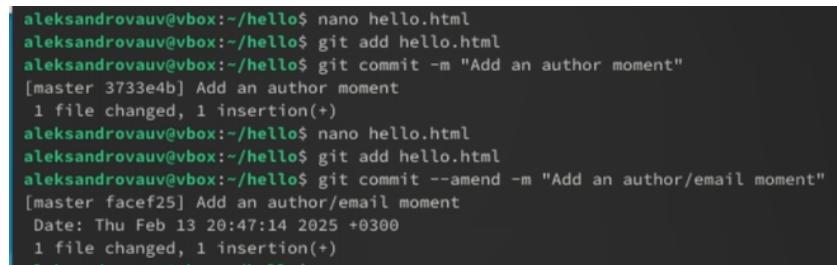


```
aleksandrova@vbox:~/hello — nano hello.html  
aleksandrova@vbox:~/hello — nano hello.html  
aleksandrova@vbox:~/hello — nano hello.html  
GNU nano 8.1  
<!-- Author: Uliana V. Aleksandrova (1132226444@pfur.ru) -->  
<html>  
  <head>  
  </head>  
  <body>  
    <h1> Hello, World :) </h1>  
  </body>  
</html>
```

Рис. 3.25: Добавление почты

### 3.8.3 1.10.3 Измените предыдущий коммит

Коммичу (рис. 3.26).



```
aleksandrova@vbox:~/hello$ nano hello.html  
aleksandrova@vbox:~/hello$ git add hello.html  
aleksandrova@vbox:~/hello$ git commit -m "Add an author moment"  
[master 3733e4b] Add an author moment  
 1 file changed, 1 insertion(+)  
aleksandrova@vbox:~/hello$ nano hello.html  
aleksandrova@vbox:~/hello$ git add hello.html  
aleksandrova@vbox:~/hello$ git commit --amend -m "Add an author/email moment"  
[master facef25] Add an author/email moment  
  Date: Thu Feb 13 20:47:14 2025 +0300  
 1 file changed, 1 insertion(+)
```

Рис. 3.26: Изменение предыдущего коммита

### 3.8.4 1.10.4 Просмотр истории

Выполняю (рис. 3.23).

```
aleksandrovauv@vbox:~/hello$ git log
commit facef25e58747c5cec3dbbe099d78b663b606037 (HEAD -> master)
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Thu Feb 13 20:47:14 2025 +0300

        Add an author/email moment
```

Рис. 3.27: Лог после изменения коммита

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

## 3.9 1.11 Перемещение файлов

### 3.9.1 1.11.1 Перемещение в каталог lib

Создаю новый репозиторий и коммичу изменения (рис. 3.28).

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

aleksandrovauv@vbox:~/hello$ git commit -m "MOVED file to lib"
[master a716d5e] MOVED file to lib
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename hello.html => lib/hello.html (100%)
```

Рис. 3.28: Каталог lib

## 3.10 1.13 Подробнее о структуре

### 3.10.1 1.13.1 Добавление index.html

Добавим файл index.html в наш репозиторий.

```
<html>
<body>
```

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

Добавляю файл и делаю коммит (рис. 3.29).

```
aleksandrova@vbox:~/hello$ touch index.html  
aleksandrova@vbox:~/hello$ nano index.html  
aleksandrova@vbox:~/hello$ git add index.html  
aleksandrova@vbox:~/hello$ git commit -m "Added index.html"  
[master 8de7879] Added index.html  
1 file changed, 5 insertions(+)  
create mode 100644 index.html  
aleksandrova@vbox:~/hello$
```

Рис. 3.29: index.html

Теперь при открытии index.html, вы должны увидеть кусок страницы hello в маленьком окошке.

### 3.10.2 1.14 Git внутри: Каталог .git

#### 3.10.2.1 1.14.1 Каталог .git

Выполняю команду (рис. 3.23):

```
ls -C .git
```

Это каталог, в котором хранится вся информация git.

```
ls -C .git/objects
```

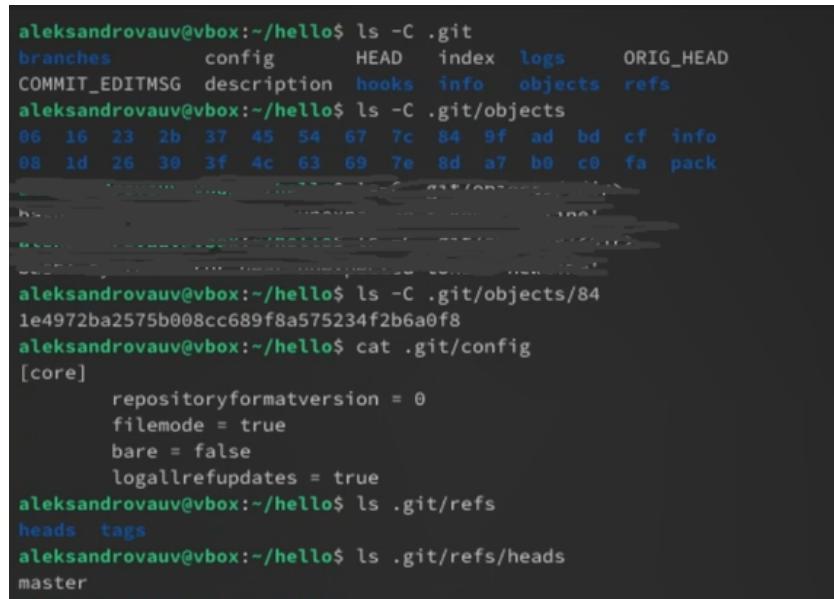
Мы видим набор каталогов, имена которых состоят из 2 символов. Имена каталогов являются первыми двумя буквами хэша sha1 объекта, хранящегося в git.

```
ls -C .git/objects/<dir>
```

Смотрю в один из каталогов с именем из 2 букв. Мы видим файлы с именами из 38 символов. Это файлы, содержащие объекты, хранящиеся в git. Они сжаты и закодированы, поэтому просмотр их содержимого нам мало чем поможет.

```
cat .git/config
```

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



```
aleksandroauv@vbox:~/hello$ ls -C .git
branches      config      HEAD      index  logs      ORIG_HEAD
COMMIT_EDITMSG description  hooks     info    objects   refs
aleksandroauv@vbox:~/hello$ ls -C .git/objects
06 16 23 2b 37 45 54 67 7c 84 9f ad bd cf info
08 1d 26 30 3f 4c 63 69 7e 8d a7 b0 c0 fa pack
aleksandroauv@vbox:~/hello$ ls -C .git/refs
heads  tags
aleksandroauv@vbox:~/hello$ ls .git/refs/heads
master
aleksandroauv@vbox:~/hello$ cat .git/config
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
aleksandroauv@vbox:~/hello$ ls .git/refs
heads  tags
aleksandroauv@vbox:~/hello$ ls .git/refs/heads
master
```

Рис. 3.30: Внутренний каталог

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

Я узнаю файлы в подкаталоге тегов. Каждый файл соответствует тегу, ранее созданному с помощью команды `git tag`. Его содержание – это всего лишь хэш коммита, привязанный к тегу. Каталог `heads` практически аналогичен, но используется для веток, а не тегов. На данный момент у нас есть только одна ветка, так что все, что мы видим в этом каталоге – это ветка `master` (рис. 3.31).

```
aleksandrova@vbox:~/hello$ ls .git/refs/heads
master
aleksandrova@vbox:~/hello$ ls .git/refs/tags
v1  v1-beta
aleksandrova@vbox:~/hello$ ls .git/refs/tags/v1
.git/refs/tags/v1
aleksandrova@vbox:~/hello$ cat .git/refs/tags/v1
c05d23f9940fec5660d3ea0426b5a7163cf43a21
aleksandrova@vbox:~/hello$ cat .git/HEAD
ref: refs/heads/master
aleksandrova@vbox:~/hello$ git log --max-count=1
commit 8de7879fe7cfb0b7e06bd6544ce41e1434cb0472 (HEAD -> master)
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Thu Feb 13 20:54:28 2025 +0300

        Added index.html
```

Рис. 3.31: сае

Файл HEAD содержит ссылку на текущую ветку, в данный момент это ветка master.

## 3.11 1.15 Работа непосредственно с объектами git

### 3.11.1 1.15.2 Вывод последнего коммита с помощью SHA1 хэша

Ищу последний коммит (рис. 3.32).

```
aleksandrova@vbox:~/hello$ git cat-file -t 8de7879fe7cfb
commit
aleksandrova@vbox:~/hello$ git cat-file -p 8de7879fe7cfb
tree 1d51ffb35d2f87c59612fdfea85be8310628b04d
parent a716d5e5a5f09e568c9bc1a09916027bbfa69268
author AleksandrovaUV <nearrmee@gmail.com> 1739469268 +0300
committer AleksandrovaUV <nearrmee@gmail.com> 1739469268 +0300

Added index.html
aleksandrova@vbox:~/hello$ git cat-file -p 1d51ffb35d2f87
100644 blob cf5bc0ea78289c0be141b578ca407f1159bd25bc    index.html
040000 tree 164b6dc67f448ea48db572beef1383cffb3651c9    lib
aleksandrova@vbox:~/hello$ git cat-file -p 164b6dc67f448ea
100644 blob 26a4dbc2208c5ef69a02d20dc89190d2edd42690    hello.html
aleksandrova@vbox:~/hello$ git cat-file -p 26a4dbc2208c5ef69a02
<!-- Author: Uliana V. Aleksandrova (1132226444@pfur.ru)-->
<html>
  <head>
  </head>
  <body>
    <h1> Hello, World :) </h1>
  </body>
</html>
```

Рис. 3.32: работа с хэш-ключами

Мне удалось найти оригинальный файл `hello.html` с самого первого коммита вручную по ссылкам SHA1 хэша в последнем коммите.

## 3.12 1.16 Создание ветки

Пора сделать наш `hello world` более выразительным. Так как это может занять некоторое время, лучше переместить эти изменения в отдельную ветку, чтобы изолировать их от изменений в ветке `master` (рис. 3.33).

```
aleksandrova@vbox:~/hello$ git checkout -b style
Switched to a new branch 'style'
aleksandrova@vbox:~/hello$ git status
On branch style
nothing to commit, working tree clean
aleksandrova@vbox:~/hello$ touch lib/style.css
aleksandrova@vbox:~/hello$ █
```

Рис. 3.33: переключение

### **3.12.1 1.16.2 Добавьте файл стилей style.css**

Создаю файл lib/style.css:

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

И выполняю:

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

### **3.12.2 1.16.3 Изменение основной страницы**

Обновляю файл hello.html, чтобы использовать стили style.css (рис. 3.34) (рис. 3.35).

```
<!-- Author: Uliana V. Aleksandrova (1132226444@pfur.ru)-->  
<html>  
    <head>  
        <link type="text/css" rel="stylesheet" media="all" href="style.css" />  
    </head>  
    <body>  
        <h1> Hello, World :) </h1>  
    </body>  
</html>
```

Рис. 3.34: Изменение файла

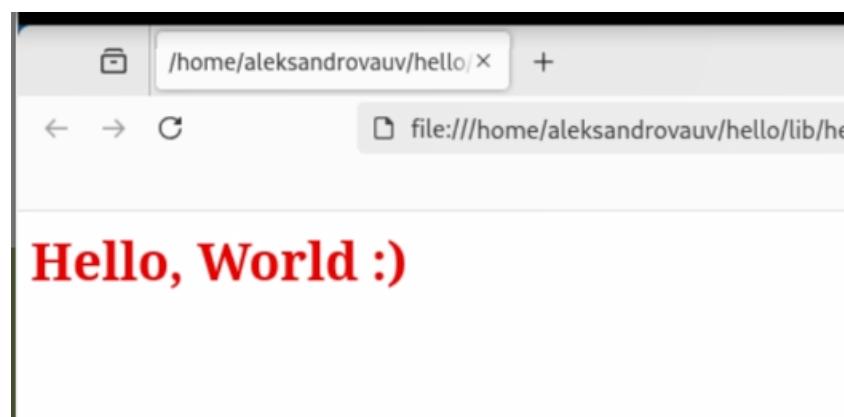


Рис. 3.35: Измененная страничка

Произвожу необходимые коммиты (рис. 3.36).

```
aleksandrova@vbox:~/hello/lib$ git add style.css
aleksandrova@vbox:~/hello/lib$ git commit -m 'Added css stylesheet'
[style 54c426d] Added css stylesheet
 1 file changed, 6 insertions(+)
  create mode 100644 lib/style.css
aleksandrova@vbox:~/hello/lib$ nano hello.html
aleksandrova@vbox:~/hello/lib$ git add hello.html
aleksandrova@vbox:~/hello/lib$ git commit -m 'Hello uses style.css'
[style fd82f76] Hello uses style.css
 1 file changed, 1 insertion(+)
```

Рис. 3.36: Коммиты

### 3.12.3 1.16.4 Изменение index.html

Обновляю файл index.html, чтобы он тоже использовал style.css.

```
<html>
  <head>
    <link type="text/css" rel="stylesheet" media="all" href="lib/style.css" />
  </head>
  <body>
    <iframe src="lib/hello.html" width="200" height="200"></iframe>
  </body>
</html>
```

И выполняю:

```
git add index.html
git commit -m "Updated index.html"
```

### 3.13 1.17 Навигация по веткам

Просмотрю каталог (рис. 3.37).

```
commit c88b2975d32c111e27e92827f751e55dc7ee9224 (HEAD -> style)
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Fri Feb 14 15:31:43 2025 +0300

    Updated index.html

commit fd82f76553e4d90b90635ff8c8b8875ee4939a0f
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Fri Feb 14 15:28:37 2025 +0300

    Hello uses style.css

commit 54c426debbec5f445efd8457803308655314a2c1
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Fri Feb 14 15:25:01 2025 +0300

    Added css stylesheet

commit 8de7879fe7cfb0b7e06bd6544ce41e1434cb0472 (master)
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Thu Feb 13 20:54:28 2025 +0300

    Added index.html
```

Рис. 3.37: Каталог

### 3.13.1 1.17.1 Переключение между ветками

Сейчас мы находимся на ветке `master`. Это заметно по тому, что файл `hello.html` не использует стили `style.css`.

Переключимся обратно. содержимое `lib/hello.html` подтверждает, что мы вернулись на ветку `style` (рис. 3.38).

```
aleksandrova@vbox:~/hello$ git checkout master
Switched to branch 'master'
aleksandrova@vbox:~/hello$ cat lib/hello.html
<!-- Author: Uliana V. Aleksandrova (1132226444@pfur.ru)-->
<html>
  <head>
  </head>
  <body>
    <h1> Hello, World :) </h1>
  </body>
</html>
aleksandrova@vbox:~/hello$ git checkout style
Switched to branch 'style'
aleksandrova@vbox:~/hello$ cat lib/hello.html
<!-- Author: Uliana V. Aleksandrova (1132226444@pfur.ru)-->
<html>
  <head>
    <link type="text/css" rel="stylesheet" media="all" href="style.css" />
  </head>
  <body>
    <h1> Hello, World :) </h1>
  </body>
</html>
```

Рис. 3.38: Переключение между ветками

##.18 Изменения в ветке master

### 3.13.2 1.18.1 Файл README в ветке master

Создаю файл README.md (рис. 3.39).

```
aleksandrova@vbox:~/hello$ git checkout master
Switched to branch 'master'
aleksandrova@vbox:~/hello$ echo "This is the Hello World example from the tutorial." >
README.md
aleksandrova@vbox:~/hello$ cat README.md
This is the Hello World example from the tutorial.
aleksandrova@vbox:~/hello$ git add README.md
aleksandrova@vbox:~/hello$ git commit -m "Added README"
[master 3a2562a] Added README
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

Рис. 3.39: Создание README

### 3.13.3 1.19.1 Просмотр отличающихся веток

### 3.13.4 1.19.2 Текущие ветки

Теперь у нас в репозитории есть две отличающиеся ветки (рис. 3.40).

```
aleksandrova@vbox:~/hello$ git log --graph --all
* commit 3a2562a3fbf595c2461ecdf5790c2da00946e2ed (HEAD -> master)
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:35:45 2025 +0300
|
|     Added README
|
* commit c88b2975d32c111e27e92827f751e55dc7ee9224 (style)
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:31:43 2025 +0300
|
|     Updated index.html
|
* commit fd82f76553e4d90b90635ff8c8b8875ee4939a0f
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:28:37 2025 +0300
|
|     Hello uses style.css
|
* commit 54c426debbec5f445efd8457803308655314a2c1
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:25:01 2025 +0300
|
|     Added css stylesheet
```

Рис. 3.40: Новая ветка

Добавление опции `--graph` в `git log` вызывает построение дерева коммитов с помощью простых ASCII символов. Мы видим обе ветки (`style` и `master`), и то, что ветка `master` является текущей `HEAD`. Общим предшественником обеих веток является коммит «Added index.html». Опция `--all` гарантированно означает, что мы видим все ветки. По умолчанию показывается только текущая ветка.

## 3.14 1.20 Слияние

### 3.14.1 1.20.1 Слияние веток

Слияние переносит изменения из двух веток в одну. Сольем `master` с `style` (рис. 3.41) (рис. 3.42).

```
aleksandrovauv@vbox:~/hello$ git checkout style
Switched to branch 'style'
aleksandrovauv@vbox:~/hello$ git merge master
Merge made by the 'ort' strategy.
 README.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

Рис. 3.41: Слияние веток

```
* commit 426db61dc8e01fd2573e5436b8db3d210d7c5bfc (HEAD -> style)
| Merge: c88b297 3a2562a
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:39:56 2025 +0300
|
|       Merge branch 'master' into style
|
* commit 3a2562a3fbf595c2461ecdf5790c2da00946e2ed (master)
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:35:45 2025 +0300
|
|       Added README
|
* commit c88b2975d32c11e27e92827f751e55dc7ee9224
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:31:43 2025 +0300
|
|       Updated index.html
|
* commit fd82f76553e4d90b90635ff8c8b8875ee4939a0f
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:28:37 2025 +0300
|
|       Hello uses style.css
|
* commit 54c426debbec5f445efd8457803308655314a2c1
| Author: AleksandrovaUV <nearrmee@gmail.com>
|/:
```

Рис. 3.42: Слияние веток лог

## 3.15 1.21 Создание конфликта

### 3.15.1 1.21.1 Вернитесь в master и создайте конфликт

Вернитесь в ветку master и внесу изменения в документ (рис. 3.43).

```
GNU nano 8.1          hello.html
<!-- Author: Uliana V. Aleksandrova (1132226444@pfur.ru)-->
<html>
  <head>
    <!-- no style-->
  </head>
  <body>
    <h1> Hello, World :) LIFE IS GREAT </h1>
  </body>
</html>
```

Рис. 3.43: Изменение файла

Отправлю изменения:

```
git add lib/hello.html
git commit -m 'Life is great'
```

### 3.15.2 1.21.2 Просмотр веток

После коммита «Added README» ветка master была объединена с веткой style, но в настоящее время в master есть дополнительный коммит, который не был слит с style. Последнее изменение в master конфликтует с некоторыми изменениями в style (рис. 3.44).

```
* commit 25dcb4ef073b59cef3d30936be04f66cdf7329b (HEAD -> master)
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:42:39 2025 +0300

|   Life is great

* commit 426db61dc8e01fd2573e5436b8db3d210d7c5bfc (style)
| Merge: c88b297 3a2562a
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:39:56 2025 +0300

|       Merge branch 'master' into style

* commit 3a2562a3fbf595c2461ecdf5790c2da00946e2ed
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:35:45 2025 +0300

|       Added README

* commit c88b2975d32c111e27e92827f751e55dc7[e9224]
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:31:43 2025 +0300

|       Updated index.html

* commit fd82f76553e4d90b90635ff8c8b8875ee4939a0f
| Author: AleksandrovaUV <nearrmee@gmail.com>
```

Рис. 3.44: Конфликт веток

## 3.16 1.22 Разрешение конфликтов

### 3.16.1 1.22.1 Слияние master с веткой style

Теперь вернусь к ветке `style` и попытаюсь объединить ее с новой веткой `master`.

```
git checkout style
git merge master
```

В файле `lib/hello.html` мы видим (рис. 3.45).

```
GNU nano 8.1                               hello.html
<!-- Author: Uliana V. Aleksandrova (1132226444@pfur.ru)-->
<html>
    <head>
<<<<< HEAD
    <link type="text/css" rel="stylesheet" media="all" href="style.css" />
=====
    <!-- no style-->
>>>>> master
    </head>
    <body>
        <h1> Hello, World :) LIFE IS GREAT </h1>
    </body>
</html>
```

Рис. 3.45: Файл

Первый раздел – версия текущей ветки (style). Второй раздел – версия ветки master.

### 3.16.2 1.22.2 Решение конфликта

Для решения конфликта, вношу изменения (рис. 3.46).

```
GNU nano 8.1                               hello.html
<!-- Author: Uliana V. Aleksandrova (1132226444@pfur.ru)-->
<html>
    <head>
        <link type="text/css" rel="stylesheet" media="all" href="style.css" />
    </head>
    <body>
        <h1> Hello, World :) LIFE IS GREAT </h1>
    </body>
</html>
```

Рис. 3.46: Изменение файла

И выполняю коммит:

```
git add lib/hello.html
git commit -m "Merged master fixed conflict."
```

## 3.17 1.23 Сброс ветки style

### 3.17.1 1.23.1 Сброс ветки style

Вернемся на ветку `style` к точке перед тем, как мы слили ее с веткой `master`.

Мы можем сбросить ветку к любому коммиту. По сути, это изменение указателя ветки на любую точку дерева коммитов. В этом случае мы хотим вернуться в ветку `style` в точку перед слиянием с `master`. Нам необходимо найти последний коммит перед слиянием.

Мы видим, что коммит «Updated index.html» был последним на ветке `style` перед слиянием. Сбросим ветку `style` к этому коммиту (рис. 3.47).

```
aleksandrovauv@vbox:~/hello$ git reset --hard c88b2975d32c111e27
HEAD is now at c88b297 Updated index.html
aleksandrovauv@vbox:~/hello$ git log --graph --all
* commit 25dcbaef073b59cef3d30936be04f66cdf7329b (master)
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:42:39 2025 +0300
|
|       Life is great
|
* commit 3a2562a3fbf595c2461ecdf5790c2da00946e2ed
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:35:45 2025 +0300
|
|       Added README
|
* commit c88b2975d32c111e27e92827f751e55dc7ee9224 (HEAD -> style)
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:31:43 2025 +0300
|
|       Updated index.html
|
* commit fd82f76553e4d90b90635ff8c8b8875ee4939a0f
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:28:37 2025 +0300
|
|       Hello uses style.css
```

Рис. 3.47: Сброс ветки

## 3.18 1.24 Сброс ветки master

### 3.18.1 1.24.1 Сброс ветки master

Добавив интерактивный режим в ветку `master`, мы внесли изменения, конфликтующие с изменениями в ветке `style`. Вернусь в ветку `master` в точку перед внесением конфликтующих изменений (рис. 3.48).

```
aleksandroauv@vbox:~/hello$ git checkout master
Switched to branch 'master'
aleksandroauv@vbox:~/hello$ git log --graph
* commit 25dcb4ef073b59cef3d30936be04f66cdf7329b (HEAD -> master)
| Author: AleksandroUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:42:39 2025 +0300
|
|       Life is great
|
* commit 3a2562a3fbf595c2461ecdf5790c2da00946e2ed
| Author: AleksandroUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:35:45 2025 +0300
|
|       Added README
|
* commit 8de7879fe7cfb0b7e06bd6544ce41e1434cb0472
| Author: AleksandroUV <nearrmee@gmail.com>
| Date:   Thu Feb 13 20:54:28 2025 +0300
|
|       Added index.html
|
* commit a716d5e5a5f09e568c9bc1a09916027bbfa69268
| Author: AleksandroUV <nearrmee@gmail.com>
| Date:   Thu Feb 13 20:51:55 2025 +0300
```

Рис. 3.48: Ветка master

Коммит «Added README» идет непосредственно перед коммитом конфликтующего интерактивного режима. Мы сбросим ветку `master` к коммиту «Added README» (рис. 3.49).

```
aleksandrovauv@vbox:~/hello$ git reset --hard 3a2562a3fbf595
HEAD is now at 3a2562a Added README
aleksandrovauv@vbox:~/hello$ git log --graph --all
* commit 3a2562a3fbf595c2461ecdf5790c2da00946e2ed (HEAD -> master)
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:35:45 2025 +0300
|
|     Added README
|
* commit c88b2975d32c11e27e92827f751e55dc7ee9224 (style)
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:31:43 2025 +0300
|
|     Updated index.html
|
* commit fd82f76553e4d90b90635ff8c8b875ee4939a0f
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:28:37 2025 +0300
|
|     Hello uses style.css
```

Рис. 3.49: Сброс ветки master

Лог выглядит, как будто репозиторий был перемотан назад во времени к точке до какого-либо слияния.

## 3.19 1.25 Перебазирование

Используем команду `rebase` вместо команды `merge` (рис. 3.50).

```
Switched to branch 'style'
aleksandrovaув@vbox:~/hello$ git rebase master
Successfully rebased and updated refs/heads/style.
aleksandrovaув@vbox:~/hello$ git log --graph
* commit f7fa1be862e185529ca5d5a143a927326316b447 (HEAD -> style)
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:31:43 2025 +0300
|
|     Updated index.html
|
* commit caf3323458acb02ab1f407ec0105dce413ab5e5d
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:28:37 2025 +0300
|
|     Hello uses style.css
|
* commit 91f62103f4b8e99e0f27098baf38f9d32c3b00da
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:25:01 2025 +0300
|
|     Added css stylesheet
|
* commit 3a2562a3fbf595c2461ecdf5790c2da00946e2ed (master)
| Author: AleksandrovaUV <nearrmee@gmail.com>
| Date:   Fri Feb 14 15:35:45 2025 +0300
|
|     Added README
```

Рис. 3.50: Перебазирование

## 3.20 1.26 Слияние в ветку master

Мы поддерживали соответствие ветки `style` с веткой `master` (с помощью `rebase`), теперь давайте солью изменения `style` в ветку `master` (рис. 3.51).

```
aleksandrovaув@vbox:~/hello$ git checkout master
Switched to branch 'master'
aleksandrovaув@vbox:~/hello$ git merge style
Updating 3a2562a..f7fa1be
Fast-forward
  index.html      | 3 +++
  lib/hello.html | 1 +
  lib/style.css  | 6 ++++++
  3 files changed, 10 insertions(+)
  create mode 100644 lib/style.css
aleksandrovaув@vbox:~/hello$ █
```

Рис. 3.51: Слияние в ветку master

## 3.21 1.27 Клонирование репозиториев

### 3.21.1 1.27.1 Клонирование репозиториев

Делаю клон своего рабочего каталога hello в домашнем репозитории (рис. ??).

```
aleksandrovaUV@vbox:~/hello$ cd ..
aleksandrovaUV@vbox:~$ pwd
/home/aleksandrovaUV
aleksandrovaUV@vbox:~$ ls
Desktop  Downloads  Music      Public          study_2024-2025_mip  Videos
Documents  hello    Pictures   study_2024-2025_mathmod  Templates
aleksandrovaUV@vbox:~$ git clone hello cloned_hello
Cloning into 'cloned_hello'...
done.
aleksandrovaUV@vbox:~$ ls
cloned_hello  Documents  hello    Pictures   study_2024-2025_mathmod  Templates
Desktop       Downloads  Music      Public          study_2024-2025_mip  Videos
aleksandrovaUV@vbox:~$ cd cloned_hello
aleksandrovaUV@vbox:~/cloned_hello$ ls
index.html  lib     README.md
```

## 3.22 1.28 Работа с клонированный репозиторием

### 3.22.1 1.28.1 Лог

Просмотрим лог (рис. 3.52).

```
aleksandrovaUV@vbox:~/cloned_hello$ git log --all
commit f7fa1be862e185529ca5d5a143a927326316b447 (HEAD -> master, origin/style, origin/master, origin/HEAD)
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Fri Feb 14 15:31:43 2025 +0300

        Updated index.html

commit caf3323458acb02ab1f407ec0105dce413ab5e5d
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Fri Feb 14 15:28:37 2025 +0300

        Hello uses style.css

commit 91f62103f4b8e99e0f27098baf38f9d32c3b00da
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Fri Feb 14 15:25:01 2025 +0300

        Added css stylesheet

commit 3a2562a3fbf595c2461ecdf5790c2d400946e2ed
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Fri Feb 14 15:35:45 2025 +0300

        Added README
```

Рис. 3.52: Лог после клонирования

Мы видим список всех коммитов в новый репозиторий.

## 3.23 1.29 Что такое origin?

Рассмотрим, что такое origin (рис. 3.53).

```
aleksandrovauv@vbox:~/cloned_hello$ git remote
origin
aleksandrovauv@vbox:~/cloned_hello$ git remote show origin
* remote origin
  Fetch URL: /home/aleksandrovauv/hello
  Push URL: /home/aleksandrovauv/hello
  HEAD branch: master
  Remote branches:
    master tracked
    style tracked
  Local branch configured for 'git pull':
    master merges with remote master
  Local ref configured for 'git push':
    master pushes to master (up to date)
aleksandrovauv@vbox:~/cloned_hello$ git branch
* master
aleksandrovauv@vbox:~/cloned_hello$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/style
aleksandrovauv@vbox:~/cloned_hello$ cd ../hello
```

Рис. 3.53: Слияние в ветку master

Удаленные репозитории обычно размещаются на отдельной машине, возможно, централизованном сервере. Однако, как мы видим здесь, они могут с тем же успехом указывать на репозиторий на той же машине.

## 3.24 1.31 Изменение оригинального репозитория

### 3.24.1 1.31.1 Внесение изменения в оригинальный репозиторий

#### hello

Сейчас мы находимся в репозитории hello. Вношу следующие изменения в файл README.md:

This is the Hello World example from the git tutorial.

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

### 3.24.2 1.31.2 Извлечение изменений

Извлекаю изменения (рис. 3.54).

```
aleksandrovauv@vbox:~/hello$ git add README.md
aleksandrovauv@vbox:~/hello$ git commit -m "Changed README in original repo"
[master caf4257] Changed README in original repo
 1 file changed, 1 insertion(+), 1 deletion(-)
aleksandrovauv@vbox:~/hello$ cd ../cloned_hello
aleksandrovauv@vbox:~/cloned_hello$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 364 bytes | 22.00 KiB/s, done.
From /home/aleksandrovauv/hello
  f7fa1be..caf4257  master      -> origin/master
aleksandrovauv@vbox:~/cloned_hello$
```

Рис. 3.54: Извлечение изменений

Мы можем продемонстрировать, что клонированный файл README.md не изменился. Поэтому я сливаю изменения в локальную ветку master. Теперь изменения есть (рис. 3.55).

```
aleksandrovauv@vbox:~/cloned_hello$ cat README.md
This is the Hello World example from the tutorial.
aleksandrovauv@vbox:~/cloned_hello$ git merge origin/master
Updating f7fa1be..caf4257
Fast-forward
 README.md | 2 ++
 1 file changed, 1 insertion(+), 1 deletion(-)
aleksandrovauv@vbox:~/cloned_hello$ cat README.md
This is the Hello World example from the git tutorial.
aleksandrovauv@vbox:~/cloned_hello$ git pull
Already up to date.
aleksandrovauv@vbox:~/cloned_hello$ git branch --track style origin/style
branch 'style' set up to track 'origin/style'.
aleksandrovauv@vbox:~/cloned_hello$ git branch -a
* master
  style
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/style
```

Рис. 3.55: Изменения

## 3.25 1.33 Добавление ветки наблюдения

### 3.25.1 1.33.1 Добавление локальной ветки, которая отслеживает удаленную

Добавляю и открываю лог (рис. 3.56).

```
aleksandrovauv@vbox:~/cloned_hello$ git log --max-count=2
commit caf4257bad04feld19333ce0cfacdiac14d2ce24 (HEAD -> master, origin/master, origin/H
EAD)
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Fri Feb 14 16:03:10 2025 +0300

    Changed README in original repo

commit f7falbe862e185529ca5d5a143a927326316b447 (origin/style, style)
Author: AleksandrovaUV <nearrmee@gmail.com>
Date:   Fri Feb 14 15:31:43 2025 +0300

    Updated index.html
aleksandrovauv@vbox:~/cloned_hello$ cd ..
aleksandrovauv@vbox:~$ git clone --bare hello hello.git
Cloning into bare repository 'hello.git'...
done.
aleksandrovauv@vbox:~$ ls hello.git
branches config description HEAD hooks info objects packed-refs refs
aleksandrovauv@vbox:~$ cd hello
aleksandrovauv@vbox:~/hello$ git remote add shared ../hello.git
aleksandrovauv@vbox:~/hello$ nano README.md
aleksandrovauv@vbox:~/hello$ cat README.md
This is the Hello World example from the git tutorial.
(Changed in the original and pushed to shared)
```

Рис. 3.56: Изменения

## 3.26 1.35 Создайте чистый репозиторий

Выполняю:

```
cd ..
git clone --bare hello hello.git
ls hello.git
```

## 3.27 1.36 Добавление удаленного репозитория

Теперь добавляю репозиторий hello.git к оригинальному репозиторию.

```
cd hello
git remote add shared ../hello.git
```

## 3.28 1.37 Отправка изменений

Редактирую файл README.md и делаю коммит.

Файл README.md:

```
This is the Hello World example from the git tutorial. (Changed in the original and pu
```

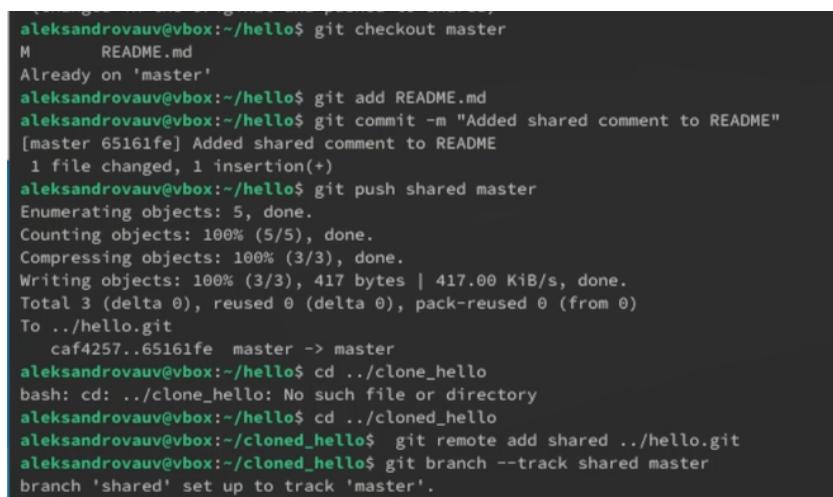
```
git checkout master
git add README
git commit -m "Added shared comment to readme"
```

Теперь отправляю изменения в общий репозиторий:

```
git push shared master
```

## 3.29 1.38 Извлечение общих изменений

Извлекаю общие изменения (рис. 3.57).



```
aleksandrova@vbox:~/hello$ git checkout master
M      README.md
Already on 'master'
aleksandrova@vbox:~/hello$ git add README.md
aleksandrova@vbox:~/hello$ git commit -m "Added shared comment to README"
[master 65161fe] Added shared comment to README
 1 file changed, 1 insertion(+)
aleksandrova@vbox:~/hello$ git push shared master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 417 bytes | 417.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To ../hello.git
  caf4257..65161fe  master -> master
aleksandrova@vbox:~/hello$ cd ../clone_hello
bash: cd: ../clone_hello: No such file or directory
aleksandrova@vbox:~/hello$ cd ../cloned_hello
aleksandrova@vbox:~/cloned_hello$ git remote add shared ../hello.git
aleksandrova@vbox:~/cloned_hello$ git branch --track shared master
branch 'shared' set up to track 'master'.
```

Рис. 3.57: Изменения

## **4 Выводы**

Я освоила работу в Git.