

Лабораторная работа №1

Работа с git

Горайнова Алёна

Содержание

1	Цель работы	7
2	Теоретическое введение	8
3	Выполнение лабораторной работы	9
3.1	Подготовка	9
3.2	Создание проекта	9
3.3	Внесение изменений	10
3.4	История	12
3.5	Получение старых версий	13
3.6	Создание тегов версий	13
3.7	Отмена локальных изменений (до индексации)	15
3.8	Отмена проиндексированных изменений (перед коммитом)	15
3.9	Отмена коммитов	16
3.10	Удаление коммитов из ветки	17
3.11	Удаление тега оорс	18
3.12	Изменение предыдущего коммита	19
3.13	Перемещение файлов	20
3.14	Подробнее о структуре	20
3.15	Git внутри: Каталог .git	21
3.16	Работа непосредственно с объектами git	22
3.17	Создание ветки	23
3.18	Навигация по веткам	24
3.19	Изменения в ветке master	26
3.20	Слияние	26
3.21	Создание конфликта	27
3.22	Разрешение конфликтов	28
3.23	Сброс ветки style	29
3.24	Сброс ветки master	31
3.25	Перебазирование	33
3.26	Слияние в ветку master	34
3.27	Клонирование репозитория	34
3.28	Что такое origin?	35
3.29	Удаленные ветки	36
3.30	Изменение оригинального репозитория	36
3.31	Слияние извлеченных изменений	37
3.32	Добавление ветки наблюдения	38

3.33 Создание чистого репозитория	38
3.34 Отправка и извлечение изменений	38
4 Выводы	40

Список иллюстраций

3.1	Настройка git	9
3.2	Создание репозитория	9
3.3	Добавление файла в репозиторий	10
3.4	Внесение изменений в содержимое репозитория	10
3.5	Внесение нескольких изменений в содержимое репозитория	11
3.6	Внесение нескольких изменений в содержимое репозитория	12
3.7	Просмотр истории	12
3.8	Просмотр разных версий репозитория	13
3.9	Создание тегов версий	14
3.10	Переключение по имени тега и просмотр доступных тегов	14
3.11	Отмена локальных изменений (до индексации)	15
3.12	Отмена проиндексированных изменений (перед коммитом)	16
3.13	Отмена коммитов	17
3.14	Удаление коммитов из ветки	18
3.15	Ничего никогда не теряется	18
3.16	Удаление тега оорс	19
3.17	Изменение предыдущего коммита	19
3.18	Изменение предыдущего коммита	20
3.19	Результат открытия index.html	21
3.20	Каталог .git	22
3.21	Работа непосредственно с объектами git	22
3.22	Создание ветки	23
3.23	Создание ветки	23
3.24	Редактирование файла	23
3.25	Редактирование файла	24
3.26	Просмотр логов новой ветки	24
3.27	Переключение на ветку master	25
3.28	Возвращение к ветке style	25
3.29	Изменения в ветке master	26
3.30	Слияние веток	27
3.31	Создание конфликта	28
3.32	Конфликт	28
3.33	Разрешение конфликта	29
3.34	Поиск коммита перед слиянием	30
3.35	Сброс ветки style	31
3.36	Поиск коммита перед конфликтом	32
3.37	Сброс ветки master	33

3.38	Перебазирование	34
3.39	Слияние style в master	34
3.40	Клонирование репозитория	35
3.41	Клонирование репозитория	35
3.42	Просмотр имени по умолчанию удаленного репозитория	36
3.43	Изменение оригинального репозитория	36
3.44	Извлечение изменений	37
3.45	Слияние извлеченных изменений	37
3.46	Добавление ветки наблюдения	38
3.47	Создание чистого репозитория	38
3.48	Отправка изменений	39
3.49	Извлечение изменений	39

Список таблиц

1 Цель работы

Приобрести практические навыки работы с системой управления версиями Git.

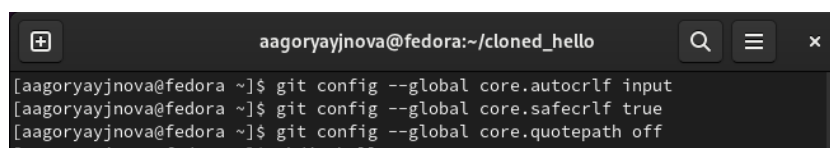
2 Теоретическое введение

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

3 Выполнение лабораторной работы

3.1 Подготовка

Сначала настроим `core.autocrlf` с параметрами `true` и `input`, чтобы сделать все переводы строк текстовых файлов в главном репозитории одинаковыми, а затем настроим отображение `unicode` (рис. fig. 3.1).

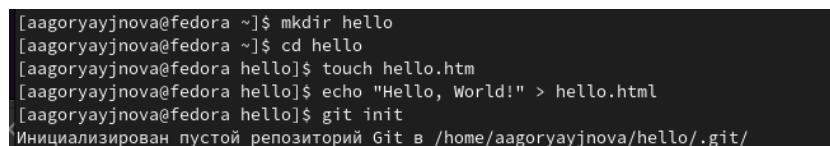
A terminal window titled 'aagoryayjnova@fedora:~/cloned_hello' with search, menu, and close icons. It shows the following commands and their outputs:

```
[aagoryayjnova@fedora ~]$ git config --global core.autocrlf input
[aagoryayjnova@fedora ~]$ git config --global core.safecrlf true
[aagoryayjnova@fedora ~]$ git config --global core.quotepath off
[aagoryayjnova@fedora ~]$
```

Рис. 3.1: Настройка git

3.2 Создание проекта

Создадим пустой каталог `hello`, а в нём файл с именем `hello.html`. Затем создадим git репозиторий из этого каталога, выполнив команду `git init`. Добавим файл в репозиторий и проверим статус, который сообщает, что коммитить нечего (рис. fig. 3.2-fig. 3.3).

A terminal window showing the following commands and their outputs:

```
[aagoryayjnova@fedora ~]$ mkdir hello
[aagoryayjnova@fedora ~]$ cd hello
[aagoryayjnova@fedora hello]$ touch hello.htm
[aagoryayjnova@fedora hello]$ echo "Hello, World!" > hello.html
[aagoryayjnova@fedora hello]$ git init
Инициализирован пустой репозиторий Git в /home/aagoryayjnova/hello/.git/
```

Рис. 3.2: Создание репозитория

```

[aagoryayjnova@fedora hello]$ git add hello.htm
[aagoryayjnova@fedora hello]$ git commit -m "Initial Commit"
[master (корневой коммит) 7869807] Initial Commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 hello.htm
[aagoryayjnova@fedora hello]$ git status
Текущая ветка: master
Неотслеживаемые файлы:
(используйте «git add <файл>...», чтобы добавить в то, что будет включено в коммит)
        hello.html
индекс пуст, но есть неотслеживаемые файлы
(используйте «git add», чтобы проиндексировать их)

```

Рис. 3.3: Добавление файла в репозиторий

3.3 Внесение изменений

Изменим содержимое файла hello.html на:

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

Проверив состояние рабочего каталога увидим, что git знает, что файл hello.html был изменен, но при этом эти изменения еще не зафиксированы в репозитории. Теперь проиндексируем изменения и снова посмотрим статус, в нём указано, что изменения пока не записаны в репозиторий. И наконец закоммитим изменения, внося их в репозиторий и снова посмотрим статус, который теперь показывает, что все изменения внесены в репозиторий (рис. fig. 3.4).

```

[aagoryayjnova@fedora hello]$ nano hello.html
[aagoryayjnova@fedora hello]$ git status
Текущая ветка: master
Изменения, которые не в индексе для коммита:
(используйте «git add/rm <файл>...», чтобы добавить или удалить файл из индекса)
(используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)
        удалено:      hello.htm
        изменено:     hello.html
индекс пуст (используйте «git add» и/или «git commit -a»)
[aagoryayjnova@fedora hello]$ git add hello.html
[aagoryayjnova@fedora hello]$ git status

```

Рис. 3.4: Внесение изменений в содержимое репозитория

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

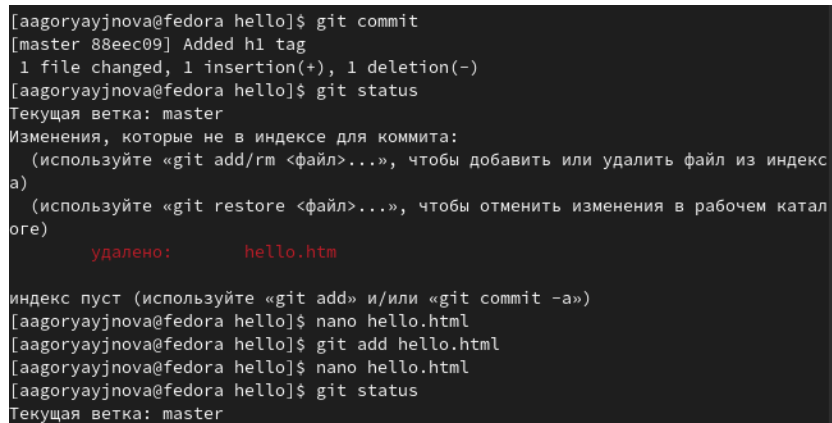
и

.

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

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

) к странице «Hello, World». Проверив текущий статус увидим, что hello.html указан дважды в состоянии. Первое изменение (добавление стандартных тегов) проиндексировано и готово к коммиту. Второе изменение (добавление заголовков HTML) является непроиндексированным. Произведем коммит проиндексированного изменения, затем проиндексируем оставшееся изменение, посмотрим статус и прокоммитим его(рис. fig. 3.5-fig. 3.6).



```
[aagoryayjnova@fedora hello]$ git commit
[master 88eec09] Added h1 tag
1 file changed, 1 insertion(+), 1 deletion(-)
[aagoryayjnova@fedora hello]$ git status
Текущая ветка: master
Изменения, которые не в индексе для коммита:
(используйте «git add/rm <файл>...», чтобы добавить или удалить файл из индекса)
(используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)

удалено:      hello.htm

индекс пуст (используйте «git add» и/или «git commit -a»)
[aagoryayjnova@fedora hello]$ nano hello.html
[aagoryayjnova@fedora hello]$ git add hello.html
[aagoryayjnova@fedora hello]$ nano hello.html
[aagoryayjnova@fedora hello]$ git status
Текущая ветка: master
```

Рис. 3.5: Внесение нескольких изменений в содержимое репозитория

```

(используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)
удалено:      hello.htm
изменено:     hello.html

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

[aagoryayjnova@fedora hello]$ git commit -m "Added HTML header"
[master 73bfd3] Added HTML header
2 files changed, 2 insertions(+)
delete mode 100644 hello.htm
[aagoryayjnova@fedora hello]$ git log
commit 73bfd367d13807dc5992f152ae9f30e819c14e7 (HEAD -> master)

```

Рис. 3.6: Внесение нескольких изменений в содержимое репозитория

3.4 История

Получим список произведённых изменений в стандартном виде, затем в однострочном, а также с указанием времени и количества (рис. fig. 3.7).

```

[aagoryayjnova@fedora hello]$ git log
commit 73bfd367d13807dc5992f152ae9f30e819c14e7 (HEAD -> master)
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:34:10 2025 +0300

    Added HTML header

commit 5af280d3776b9f4853d23a98f83b729696c766ca
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:32:28 2025 +0300

    Added standard HTML page tags

commit 88eec0984435339897f41a9b4929cd54206bbbe7
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:27:58 2025 +0300

    Added h1 tag

commit a5bc68d8462ee1240f9a44214807eb63ec91b610
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:21:01 2025 +0300

    Initial Commit

commit 7869807b53ac507d42adbcea8766057395653920
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:19:01 2025 +0300

    Initial Commit

```

Рис. 3.7: Просмотр истории

3.5 Получение старых версий

Изучим данные лога и найдем там хэш первого коммита, используя его вернемся к первой версии и просмотрим файл `hello.html`, действительно, увидим первую версию. Затем вернемся к последней версии в ветке `master` и вновь посмотрим на файл(рис. fig. 3.8).

```
[aagoryayjnova@fedora hello]$ git checkout ^C
[aagoryayjnova@fedora hello]$ git checkout 7869807b53a
Примечание: переключение на «7869807b53a».

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 7869807 Initial Commit
[aagoryayjnova@fedora hello]$ cat hello.html
cat: hello.html: Нет такого файла или каталога
[aagoryayjnova@fedora hello]$ git checkout 88eec0984435339
Предыдущая позиция HEAD была 7869807 Initial Commit
HEAD сейчас на 88eec09 Added h1 tag
[aagoryayjnova@fedora hello]$ cat hello.html
<h1>Hello World</h1>
```

Рис. 3.8: Просмотр разных версий репозитория

3.6 Создание тегов версий

Назовем текущую версию страницы `hello` первой (`v1`). Создадим тег первой версии и используем его для того чтобы вернуться к предыдущей, которой также присвоим тег(рис. fig. 3.9).

```
[aagoryayjnova@fedora hello]$ git tag v1
[aagoryayjnova@fedora hello]$ git checkout v1^
Примечание: переключение на «v1^».
```

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

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

```
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 5af280d Added standard HTML page tags

```
[aagoryayjnova@fedora hello]$ cat hello.html
<html>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

```
[aagoryayjnova@fedora hello]$ git tag v1-beta
```

Рис. 3.9: Создание тегов версий

Переключимся по тегам между двумя отмеченными версиями. Просмотрим все доступные теги(их два) и посмотрим теги в логе(рис. fig. 3.10).

```
[aagoryayjnova@fedora hello]$ git log master --all
commit 73bffd367d13807dc5992f152ae9f30e819c14e7 (tag: v1, master)
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:34:10 2025 +0300

    Added HTML header

commit 5af280d3776b9f4853d23a98f83b729696c766ca (HEAD, tag: v1-beta)
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:32:28 2025 +0300

    Added standard HTML page tags

commit 88eec0984435339897f41a9b4929cd54206bbbe7
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:27:58 2025 +0300

    Added h1 tag

commit a5bc68d8462ee1240f9a44214807eb63ec91b610
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:21:01 2025 +0300

    Initial Commit

commit 7869807b53ac507d42adbcea8766057395653920
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:14:12 2025 +0300

    Initial Commit
```

Рис. 3.10: Переключение по имени тега и просмотр доступных тегов

3.7 Отмена локальных изменений (до индексации)

Убедимся, что мы находимся на последнем коммите ветки `master` и внесем изменение в файл `hello.html` в виде нежелательного комментария. Затем проверим статус, увидим, что изменения ещё не проиндексированы. Используем команду `git checkout` для переключения версии файла `hello.html` в репозитории (рис. fig. 3.11).

```
[aagoryayjnova@fedora hello]$ git checkout master
Предыдущая позиция HEAD была 5af280d Added standard HTML page tags
Переключились на ветку «master»
[aagoryayjnova@fedora hello]$ git checkout master
Уже на «master»
[aagoryayjnova@fedora hello]$ nano hello.html
[aagoryayjnova@fedora 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»)
[aagoryayjnova@fedora hello]$ git checkout hello.html
Updated 1 path from the index
[aagoryayjnova@fedora hello]$ git status
Текущая ветка: master
ничего коммитить, нет изменений в рабочем каталоге
[aagoryayjnova@fedora hello]$ cat hello.html
<html>
<head>
</head>
<body>
  <h1>Hello, World!</h1>
</body>
</html>
[aagoryayjnova@fedora hello]$ nano hello.html
```

Рис. 3.11: Отмена локальных изменений (до индексации)

3.8 Отмена проиндексированных изменений (перед коммитом)

Внесем изменение в файл `hello.html` в виде нежелательного комментария

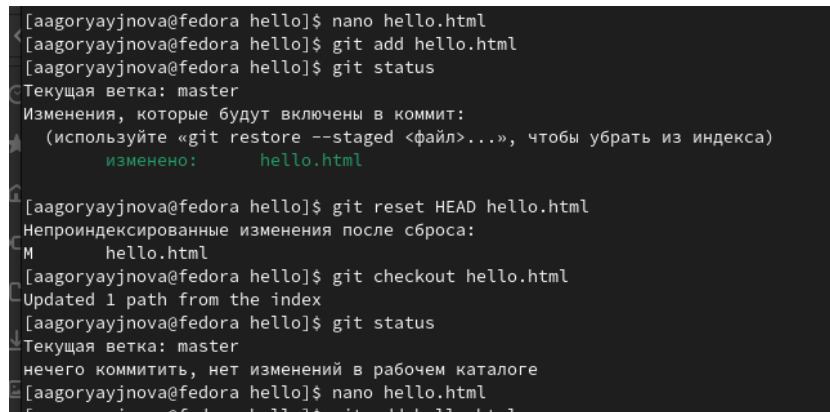
```
<html>
<head>
  <!-- This is an unwanted but staged comment -->
```

```

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

```

Проиндексируем это изменение и проверим состояние. Состояние показывает, что изменение было проиндексировано и готово к коммиту. Используем команду `git reset`, чтобы сбросить буферную зону к HEAD. Это очищает буферную зону от изменений, которые мы только что проиндексировали. И переключимся на последнюю версию коммита, посмотрев статус увидим, что наш каталог опять чист(рис. fig. 3.12).



```

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

[aagoryayjnova@fedora hello]$ git reset HEAD hello.html
Непроиндексированные изменения после сброса:
M       hello.html
[aagoryayjnova@fedora hello]$ git checkout hello.html
Updated 1 path from the index
[aagoryayjnova@fedora hello]$ git status
Текущая ветка: master
ничего коммитить, нет изменений в рабочем каталоге
[aagoryayjnova@fedora hello]$ nano hello.html
[aagoryayjnova@fedora hello]$ git add hello.html

```

Рис. 3.12: Отмена проиндексированных изменений (перед коммитом)

3.9 Отмена коммитов

Изменим файл `hello.html` на следующий.

```

<html>
  <head>
  </head>
  <body>

```



```

<h1>Hello, World!</h1>

<!-- This is an unwanted but committed change -->

</body>
</html>

```

Проиндексируем изменения файла и прокоммитим их. Чтобы отменить коммит, нам необходимо сделать коммит, который удаляет изменения, сохраненные нежелательным коммитом. Перейдем в редактор, где изменим нежелательный коммит. Проверим лог. Проверка лога показывает нежелательные и отмененные коммиты в наш репозиторий(рис. fig. 3.13).

```

[aagoryayjnova@fedora hello]$ nano hello.html
[aagoryayjnova@fedora hello]$ git add hello.html
[aagoryayjnova@fedora hello]$ git commit -m "Oops, we didn't want this commit"
[master 8cfa8cd] Oops, we didn't want this commit
1 file changed, 1 insertion(+)
[aagoryayjnova@fedora hello]$ git revert HEAD
[master f755d76] Revert "Oops, we didn't want this commit"
1 file changed, 1 deletion(-)
[aagoryayjnova@fedora hello]$ git log
commit f755d76327c48c52fdd4b4cd6e19d2c8f993f442 (HEAD -> master)
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 21:04:42 2025 +0300

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

    This reverts commit 8cfa8cd7666a2fe2608d9c17146a03f4280f6897.

commit 8cfa8cd7666a2fe2608d9c17146a03f4280f6897
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 21:04:21 2025 +0300

    Oops, we didn't want this commit

```

Рис. 3.13: Отмена коммитов

3.10 Удаление коммитов из ветки

Удалим последние два коммита с помощью сброса, сначала отметим последний коммит тегом, чтобы его можно было потом найти. Используем команду `git reset`, чтобы вернуться к версии до этих коммитов. Теперь в логе их нет, но если посмотреть логи с опцией `-all` можно всё ещё их увидеть, но метка HEAD находится на нужной нам версии(рис. fig. 3.14-fig. 3.15).

```

[aaagoryayjnova@fedora hello]$ git tag oops
[aaagoryayjnova@fedora hello]$ git reset --hard v1
Указатель HEAD сейчас на коммите 73bfd3 Added HTML header
[aaagoryayjnova@fedora hello]$ git log
commit 73bfd367d13807dc5992f152ae9f30e819c14e7 (HEAD -> master, tag: v1)
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:34:10 2025 +0300

    Added HTML header

commit 5af280d3776b9f4853d23a98f83b729696c766ca (tag: v1-beta)
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:32:28 2025 +0300

    Added standard HTML page tags

commit 88eec0984435339897f41a9b4929cd54206bbe7
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:27:58 2025 +0300

    Added h1 tag

```

Рис. 3.14: Удаление коммитов из ветки

```

метка «oops» удалена (была 73bfd3)
[aaagoryayjnova@fedora hello]$ git log --all
commit 73bfd367d13807dc5992f152ae9f30e819c14e7 (HEAD -> master, tag: v1)
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:34:10 2025 +0300

    Added HTML header

commit 5af280d3776b9f4853d23a98f83b729696c766ca (tag: v1-beta)
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:32:28 2025 +0300

    Added standard HTML page tags

commit 88eec0984435339897f41a9b4929cd54206bbe7
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:27:58 2025 +0300

    Added h1 tag

commit a5bc68d8462ee1240f9a44214807eb63ec91b610
Author: Алёна Горяйнова <alenagor2004@mail.ru>

```

Рис. 3.15: Ничего никогда не теряется

3.11 Удаление тега оорс

Удалим тег оорс и коммиты, на которые он ссылался, сборщиком мусора. Теперь этот тег не отображается в репозитории(рис. fig. 3.16).

```
[aagoryayjnova@fedora hello]$ git tag -d oops
Метка «oops» удалена (была f755d76)
[aagoryayjnova@fedora hello]$ git log --all
commit 73bf1d367d13807dc5992f152ae9f30e819c14e7 (HEAD -> master, tag: v1)
Author: Алёна Горайнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:34:10 2025 +0300

    Added HTML header

commit 5af280d3776b9f4853d23a98f83b729696c766ca (tag: v1-beta)
Author: Алёна Горайнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:32:28 2025 +0300

    Added standard HTML page tags

commit 88eec0984435339897f41a9b4929cd54206bbbe7
Author: Алёна Горайнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:27:58 2025 +0300

    Added h1 tag
```

Рис. 3.16: Удаление тега oops

3.12 Изменение предыдущего коммита

Добавим в страницу комментарий автора (рис. fig. 3.17).

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

Рис. 3.17: Изменение предыдущего коммита

Затем добавим их в репозиторий. Теперь мы хотим добавить в комментарий автора почту, обновиим страницу hello, включив в неё почту. Чтобы у нас остался один коммит, а не два, изменим последний с помощью опции `-amend`, теперь в логах отображается последняя версия коммита(рис. fig. 3.18).

```

[aagoryayjnova@fedora hello]$ git add hello.html
[aagoryayjnova@fedora hello]$ git commit -m "Add an author comment"
[master 2020aee] Add an author comment
1 file changed, 1 insertion(+)
[aagoryayjnova@fedora hello]$ nano hello.html
[aagoryayjnova@fedora hello]$ git add hello.html
[aagoryayjnova@fedora hello]$ git commit --amend -m "Add an author/email comment"
[master 9b85f00] Add an author/email comment
Date: Thu Feb 20 21:11:08 2025 +0300
1 file changed, 1 insertion(+)
[aagoryayjnova@fedora hello]$ git log
commit 9b85f00eadc985f75bdd5ad807d2015fb2d6b4cc (HEAD -> master)
Author: Алёна Горайнова <alenagor2004@mail.ru>
Date: Thu Feb 20 21:11:08 2025 +0300

    Add an author/email comment

commit 73bf1d367d13807dc5992f152ae9f30e819c14e7 (tag: v1)
Author: Алёна Горайнова <alenagor2004@mail.ru>
Date: Thu Feb 20 20:34:10 2025 +0300

    Added HTML header

```

Рис. 3.18: Изменение предыдущего коммита

3.13 Перемещение файлов

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

3.14 Подробнее о структуре

Добавим файл `index.html` в наш репозиторий

```

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

```

Добавим файл и сделаем коммит.

Теперь при открытии `index.html`, увидим кусок страницы `hello` в маленьком окошке(рис.@fig:019).

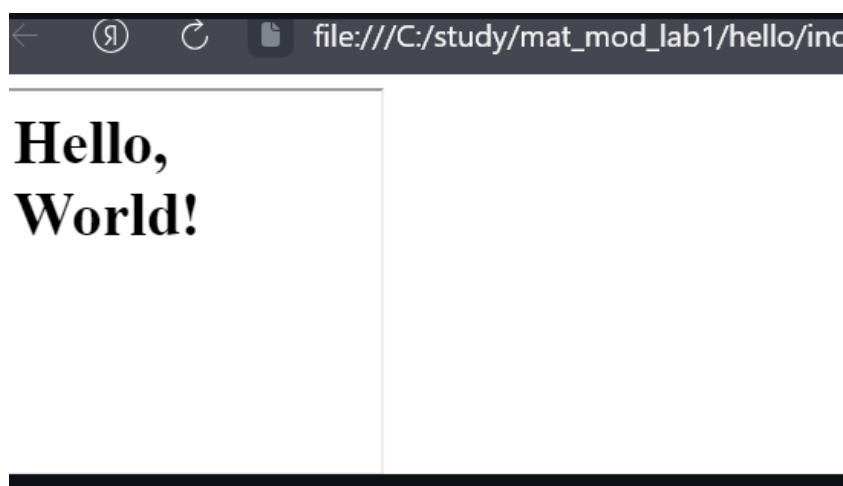


Рис. 3.19: Результат открытия index.html

3.15 Git внутри: Каталог .git

Просмотрим каталог, в котором хранится вся информация git. Затем посмотрим набор каталогов, имена которых состоят из 2 символов. Имена каталогов являются первыми двумя буквами хэша sha1 объекта, хранящегося в git. Посмотрим в один из каталогов с именем из 2 букв. Увидим файлы с именами из 38 символов. Это файлы, содержащие объекты, хранящиеся в git. Посмотрим файл конфигурации, создающийся для каждого конкретного проекта. Затем посмотрим подкаталоги `.git/refs/heads` и `.git/refs/tags`, а также содержимое файла `v1`, в нём хранится хэш коммита, привязанный к тегу. Также посмотрим содержимое файла `HEAD`, который содержит ссылку на текущую ветку, в данный момент это ветка `master`(рис. fig. 3.20).

```

create mode 100644 index.html
[aagoryayjnova@fedora hello]$ ls -C .git
branches      config        HEAD   index  logs    ORIG_HEAD  refs
COMMIT_EDITMSG  description  hooks  info   objects packed-refs
[aagoryayjnova@fedora hello]$ ls -C .git/objects
bash: s: команда не найдена...
[aagoryayjnova@fedora hello]$ ls -C .git/objects
0d 1a 20 39 5a 69 70 78 8a 9b a9 b3 bd d5 e6 info
0e 1b 22 49 64 6d 73 88 8c a5 ae bc cd d7 f7 pack
[aagoryayjnova@fedora hello]$ ls -C .git/objects/<dir>
bash: синтаксическая ошибка рядом с неожиданным маркером «newline»
[aagoryayjnova@fedora hello]$ ls -C .git/objects/dir
ls: невозможно получить доступ к '.git/objects/dir': Нет такого файла или каталога
[aagoryayjnova@fedora hello]$ ls -C .git/objects/<dir>
bash: синтаксическая ошибка рядом с неожиданным маркером «newline»
[aagoryayjnova@fedora hello]$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
[aagoryayjnova@fedora hello]$ ls .git/refs
heads  tags
[aagoryayjnova@fedora hello]$ ls .git/refs/heads
master
[aagoryayjnova@fedora hello]$ cat .git/HEAD
ref: refs/heads/master

```

Рис. 3.20: Каталог .git

3.16 Работа непосредственно с объектами git

Найдем последний коммит и выведем его с помощью SHA1 хэша. Затем посмотрим дерево каталогов, ссылка на который идёт в последнем коммите, выведем каталог lib и файл hello.html(рис. fig. 3.21).

```

[aagoryayjnova@fedora hello]$ git log --max-count=1
commit 6490608674720819c7f1f0bc670571c55f99dc1e (HEAD -> master)
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 21:17:36 2025 +0300

    Added index.html.
[aagoryayjnova@fedora hello]$ git cat-file -t <hash>
bash: синтаксическая ошибка рядом с неожиданным маркером «newline»

```

Рис. 3.21: Работа непосредственно с объектами git

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

3.17 Создание ветки

Создадим новую ветку «style» и перейдем в неё. Добавим туда файл стилей style.css и добавим его в репозиторий. Обновим файл hello.html, чтобы использовать стили style.css и index.html, также обновим их в репозиторий(рис. fig. 3.22-fig. 3.25).

```
[aagoryayjnova@fedora hello]$ git checkout -b style~
fatal: 'style~' is not a valid branch name
[aagoryayjnova@fedora hello]$ git checkout -b style
Переключились на новую ветку «style»
[aagoryayjnova@fedora hello]$ git status
Текущая ветка: style
ничего коммитить, нет изменений в рабочем каталоге
[aagoryayjnova@fedora hello]$ touch lib/style.css
[aagoryayjnova@fedora hello]$ nano lib/style.css
[aagoryayjnova@fedora hello]$ git add lib/style.css
[aagoryayjnova@fedora hello]$ git commit -m "Added css stylesheet"
```

Рис. 3.22: Создание ветки

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

Рис. 3.23: Создание ветки

```
<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" />
</body>
</html>
```

Рис. 3.24: Редактирование файла

```

1 file changed, 4 insertions(+)
[aagoryayjnova@fedora hello]$ git log --all
commit 74dc620ae681e3150ac0912c45bc9f42abe3a0a2 (HEAD -> style)
Author: Алёна Горайнова <alenagor2004@mail.ru>
Date: Thu Feb 20 21:29:53 2025 +0300

    Updated index.html

commit d581604553d22f90b0726650d9f3e4bdcf4c5c03
Author: Алёна Горайнова <alenagor2004@mail.ru>
Date: Thu Feb 20 21:27:30 2025 +0300

    Hello uses style.css

commit 6dfaede781db0764781e0df6fff227f685039831
Author: Алёна Горайнова <alenagor2004@mail.ru>

```

Рис. 3.25: Редактирование файла

3.18 Навигация по веткам

Посмотрим все логи(рис. fig. 3.26).

```

[aagoryayjnova@fedora hello]$ git checkout master
Переключились на ветку «master»
[aagoryayjnova@fedora hello]$ cat lib/hello.html
<!-- Author: Dmitry S. Kulyabov (kulyabov-ds@rudn.ru) -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
[aagoryayjnova@fedora hello]$ git checkout style
Переключились на ветку «style»
[aagoryayjnova@fedora hello]$ cat lib/hello.html
<!-- Author: Dmitry S. Kulyabov (kulyabov-ds@rudn.ru) -->
<html>

```

Рис. 3.26: Просмотр логов новой ветки

Переключимся обратно на основную ветку и посмотрим содержимое файла `lib/hello.html`, заметим, что он не использует стили, также посмотрим содержимое этого файла в новой ветке(рис. fig. 3.27-fig. 3.28).


```

[aagoryayjnova@fedora hello]$ git checkout style
Переключились на ветку «style»
[aagoryayjnova@fedora hello]$ cat lib/hello.html
<!-- Author: Dmitry S. Kulyabov (kulyabov-ds@rudn.ru) -->
<html>
  <head>
    <link type="text/css" rel="stylesheet"
          media="all" href="style.css" />
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
[aagoryayjnova@fedora hello]$ git checkout master
Переключились на ветку «master»

```

Рис. 3.27: Переключение на ветку master

```

[aagoryayjnova@fedora hello]$ git checkout master
Переключились на ветку «master»
[aagoryayjnova@fedora hello]$ touch README.md
[aagoryayjnova@fedora hello]$ echo "This is the Hello World example from
tutorial." > README.md
[aagoryayjnova@fedora hello]$ git log --graph --all
* commit 74dc620ae681e3150ac0912c45bc9f42abe3a0a2 (style)
  Author: Алёна Горайнова <alenagor2004@mail.ru>
  Date: Thu Feb 20 21:29:53 2025 +0300

    Updated index.html

* commit d581604553d22f90b0726650d9f3e4bdcf4c5c03
  Author: Алёна Горайнова <alenagor2004@mail.ru>
  Date: Thu Feb 20 21:27:30 2025 +0300

    Hello uses style.css

* commit 6dfaede781db0764781e0df6ffff227f685039831
  Author: Алёна Горайнова <alenagor2004@mail.ru>
  Date: Thu Feb 20 21:24:49 2025 +0300

    Added css stylesheet

* commit 6490608674720819c7f1f0bc670571c55f99dc1e (HEAD -> master)
  Author: Алёна Горайнова <alenagor2004@mail.ru>
  Date: Thu Feb 20 21:17:36 2025 +0300

    Added index.html.

* commit 3969c3a835b7a9b4bf4a36f3f4fe5ad921289beb
  Author: Алёна Горайнова <alenagor2004@mail.ru>
  Date: Thu Feb 20 21:15:46 2025 +0300

    Moved hello.html to lib

* commit 9b85f00eadc985f75bdd5ad807d2015fb2d6b4cc

```

Рис. 3.28: Возвращение к ветке style

3.19 Изменения в ветке master

Вернемся в основную ветку и добавим файл README.md. Просмотрим ветки и их различия(рис. fig. 3.29).

```
:
[6]+ Остановлен  git log --graph --all
[aagoryayjnova@fedora hello]$ git checkout style
Переключились на ветку «style»
[aagoryayjnova@fedora hello]$ git merge master
Уже актуально.
[aagoryayjnova@fedora hello]$ git log --graph --all
* commit 74dc620ae681e3150ac0912c45bc9f42abe3a0a2 (HEAD -> style)
  Author: Алёна Горайнова <alenagor2004@mail.ru>
  Date:   Thu Feb 20 21:29:53 2025 +0300

    Updated index.html

* commit d581604553d22f90b0726650d9f3e4bdcf4c5c03
  Author: Алёна Горайнова <alenagor2004@mail.ru>
  Date:   Thu Feb 20 21:27:30 2025 +0300

    Hello uses style.css

* commit 6dfaede781db0764781e0df6ffff227f685039831
  Author: Алёна Горайнова <alenagor2004@mail.ru>
  Date:   Thu Feb 20 21:24:49 2025 +0300

    Added css stylesheet

* commit 6490608674720819c7f1f0bc670571c55f99dc1e (master)
  Author: Алёна Горайнова <alenagor2004@mail.ru>
  Date:   Thu Feb 20 21:17:36 2025 +0300

    Added index.html.

* commit 3969c3a835b7a9b4bf4a36f3f4fe5ad921289beb
  Author: Алёна Горайнова <alenagor2004@mail.ru>
  Date:   Thu Feb 20 21:15:46 2025 +0300

    Moved hello.html to lib
```

Рис. 3.29: Изменения в ветке master

3.20 Слияние

Слияние переносит изменения из двух веток в одну. Вернемся к ветке style и сольем master с style(рис. fig. 3.30).

```
[aagoryayjnova@fedora hello]$ nano lib/hello.html
[aagoryayjnova@fedora hello]$ git add lib/hello.html
[aagoryayjnova@fedora hello]$ git commit -m 'Life is great'
[master 8dbd09b] Life is great
1 file changed, 1 insertion(+)
[aagoryayjnova@fedora hello]$ git log --graph --all
* commit 8dbd09b302dc32480171d8fb99c98886c2f9b1d2 (HEAD -> master)
  Author: Алёна Горяйнова <alenagor2004@mail.ru>
  Date: Thu Feb 20 21:35:40 2025 +0300

    Life is great

* commit 74dc620ae681e3150ac0912c45bc9f42abe3a0a2 (style)
  Author: Алёна Горяйнова <alenagor2004@mail.ru>
  Date: Thu Feb 20 21:29:53 2025 +0300

    Updated index.html

* commit d581604553d22f90b0726650d9f3e4bdcf4c5c03
  Author: Алёна Горяйнова <alenagor2004@mail.ru>
  Date: Thu Feb 20 21:27:30 2025 +0300

    Hello uses style.css

* commit 6dfaede781db0764781e0df6fff227f685039831
  Author: Алёна Горяйнова <alenagor2004@mail.ru>
  Date: Thu Feb 20 21:24:49 2025 +0300

    Added css stylesheet

* commit 6490608674720819c7f1f0bc670571c55f99dc1e
  Author: Алёна Горяйнова <alenagor2004@mail.ru>
  Date: Thu Feb 20 21:17:36 2025 +0300

    Added index.html.

* commit 3969c3a835b7a9b4bf4a36f3f4fe5ad921289beb
```

Рис. 3.30: Слияние веток

3.21 Создание конфликта

Вернемся в ветку `master` и создадим конфликт, внося изменения в файл `hello.html`. Просмотрим ветки. После коммита «Added README» ветка `master` была объединена с веткой `style`, но в настоящее время в `master` есть дополнительный коммит, который не был слит с `style`. Последнее изменение в `master` конфликтует с некоторыми изменениями в `style`(рис. fig. 3.31).

```

<html>
  <head>
<<<<<< HEAD
    <link type="text/css" rel="stylesheet"
      media="all" href="style.css" />
=====
    <!-- no style -->
>>>>>> master
  </head>
  <body>
    <h1>hello, world!</h1>
  </body>
</html>

```

Рис. 3.31: Создание конфликта

3.22 Разрешение конфликтов

Вернемся к ветке `style` и попытаемся объединить ее с новой веткой `master`. В файле `lib/hello.html` можно увидеть записи с обеих версий этого файла. Первый раздел — версия текущей ветки (`style`). Второй раздел — версия ветки `master`. Внесем изменения в `lib/hello.html`, оставив только необходимую нам запись и добавим этот файл в репозиторий, чтобы вручную разрешить конфликт (рис. fig. 3.32-fig. 3.33).

```

<html>
  <head>
    <link type="text/css" rel="stylesheet"
      media="all" href="style.css" />
  </head>
  <body>
    <h1>hello, world!</h1>
  </body>
</html>

```

Рис. 3.32: Конфликт

```

[aagoryayjnova@fedora hello]$ git add lib/hello.html
[aagoryayjnova@fedora hello]$ git commit -m "Merged master fixed conflict."
[style 1077b37] Merged master fixed conflict.
[aagoryayjnova@fedora hello]$ git checkout style
Уже на «style»
[aagoryayjnova@fedora hello]$ git log --graph
*   commit 1077b37610f752e06ebb5c214d027e2bf84ee453 (HEAD -> style)
|\  Merge: 74dc620 8dbd09b
| | Author: Алёна Горяйнова <alenagor2004@mail.ru>
| | Date: Thu Feb 20 21:40:11 2025 +0300
| |
| | Merged master fixed conflict.
| |
*   commit 8dbd09b302dc32480171d8fb99c98886c2f9b1d2 (master)
| | Author: Алёна Горяйнова <alenagor2004@mail.ru>
| | Date: Thu Feb 20 21:35:40 2025 +0300
| |
| | Life is great
| |
*   commit 74dc620ae681e3150ac0912c45bc9f42abe3a0a2
| | Author: Алёна Горяйнова <alenagor2004@mail.ru>
| | Date: Thu Feb 20 21:29:53 2025 +0300
| |
| | Updated index.html
| |
*   commit d581604553d22f90b0726650d9f3e4bdcf4c5c03
| | Author: Алёна Горяйнова <alenagor2004@mail.ru>
| | Date: Thu Feb 20 21:27:30 2025 +0300
| |
| | Hello uses style.css
| |
*   commit 6dfaede781db0764781e0df6fff227f685039831
| | Author: Алёна Горяйнова <alenagor2004@mail.ru>
| | Date: Thu Feb 20 21:24:49 2025 +0300
| |
| | Added css stylesheet
| |
*   commit 6490608674720819c7f1f0bc670571c55f99dc1e
| | Author: Алёна Горяйнова <alenagor2004@mail.ru>
| | Date: Thu Feb 20 21:17:36 2025 +0300

```

Рис. 3.33: Разрешение конфликта

3.23 Сброс ветки style

Вернемся на ветке style к точке перед тем, как мы слили ее с веткой master. Мы хотим вернуться в ветке style в точку перед слиянием с master. Нам необходимо найти последний коммит перед слиянием(рис. fig. 3.34).

```

*   commit 1077b37610f752e06ebb5c214d027e2bf84ee453 (HEAD -> style)
| \ Merge: 74dc620 8dbd09b
| Author: Алёна Горяйнова <alenagor2004@mail.ru>
| Date: Thu Feb 20 21:40:11 2025 +0300
|
| Merged master fixed conflict.
|
*   commit 8dbd09b302dc32480171d8fb99c98886c2f9b1d2 (master)
| | Author: Алёна Горяйнова <alenagor2004@mail.ru>
| | Date: Thu Feb 20 21:35:40 2025 +0300
| |
| | Life is great
| |
*   commit 74dc620ae681e3150ac0912c45bc9f42abe3a0a2
| | Author: Алёна Горяйнова <alenagor2004@mail.ru>
| | Date: Thu Feb 20 21:29:53 2025 +0300
| |
| | Updated index.html
| |
*   commit d581604553d22f90b0726650d9f3e4bdcf4c5c03
| | Author: Алёна Горяйнова <alenagor2004@mail.ru>
| | Date: Thu Feb 20 21:27:30 2025 +0300
| |
| | Hello uses style.css
| |
*   commit 6dfaede781db0764781e0df6fff227f685039831
| | Author: Алёна Горяйнова <alenagor2004@mail.ru>
| | Date: Thu Feb 20 21:24:49 2025 +0300
| |
| | Added css stylesheet
| |
*   commit 6490608674720819c7f1f0bc670571c55f99dc1e
| | Author: Алёна Горяйнова <alenagor2004@mail.ru>
| | Date: Thu Feb 20 21:17:36 2025 +0300
| |
| |
[10]+ Остановлен git log --graph --all
[aagoryaynova@fedora hello]$ git checkout style
Уже на «style»
[aagoryaynova@fedora hello]$ git checkout master
Переключились на ветку «master»
[aagoryaynova@fedora hello]$ git merge style

```

Рис. 3.34: 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. fig. 3.35).

```

* commit 1077b37610f752e06ebb5c214d027e2bf84ee453 (HEAD -> style)
| \ Merge: 74dc620 8dbd09b
| Author: Алёна Горяйнова <alenagor2004@mail.ru>
| Date: Thu Feb 20 21:40:11 2025 +0300
|
| Merged master fixed conflict.
|
* commit 8dbd09b302dc32480171d8fb99c98886c2f9b1d2 (master)
| Author: Алёна Горяйнова <alenagor2004@mail.ru>
| Date: Thu Feb 20 21:35:40 2025 +0300
|
| Life is great
|
* commit 74dc620ae681e3150ac0912c45bc9f42abe3a0a2
| Author: Алёна Горяйнова <alenagor2004@mail.ru>
| Date: Thu Feb 20 21:29:53 2025 +0300
|
| Updated index.html
|
* commit d581604553d22f90b0726650d9f3e4bdcf4c5c03
| Author: Алёна Горяйнова <alenagor2004@mail.ru>
| Date: Thu Feb 20 21:27:30 2025 +0300
|
| Hello uses style.css
|
* commit 6dfaede781db0764781e0df6fff227f685039831
| / Author: Алёна Горяйнова <alenagor2004@mail.ru>
| Date: Thu Feb 20 21:24:49 2025 +0300
|
| Added css stylesheet
|
* commit 6490608674720819c7f1f0bc670571c55f99dc1e
| Author: Алёна Горяйнова <alenagor2004@mail.ru>
| Date: Thu Feb 20 21:17:36 2025 +0300
|
|
[10]+ Остановлен git log --graph --all
[aagoryaynova@fedora hello]$ git checkout style
Уже на «style»
[aagoryaynova@fedora hello]$ git checkout master
Переключились на ветку «master»
[aagoryaynova@fedora hello]$ git merge style

```

Рис. 3.35: Сброс ветки style

3.24 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. Посмотрим коммиты ветки master (рис. fig. 3.36).

```
* commit 1077b37610f752e06ebb5c214d027e2bf84ee453 (HEAD -> style)
| Merge: 74dc620 8dbd09b
| Author: Алёна Горяйнова <alenagor2004@mail.ru>
| Date: Thu Feb 20 21:40:11 2025 +0300
|
| Merged master fixed conflict.
|
* commit 8dbd09b302dc32480171d8fb99c98886c2f9b1d2 (master)
| Author: Алёна Горяйнова <alenagor2004@mail.ru>
| Date: Thu Feb 20 21:35:40 2025 +0300
|
| Life is great
|
* commit 74dc620ae681e3150ac0912c45bc9f42abe3a0a2
| Author: Алёна Горяйнова <alenagor2004@mail.ru>
| Date: Thu Feb 20 21:29:53 2025 +0300
|
| Updated index.html
|
* commit d581604553d22f90b0726650d9f3e4bdcf4c5c03
| Author: Алёна Горяйнова <alenagor2004@mail.ru>
| Date: Thu Feb 20 21:27:30 2025 +0300
|
| Hello uses style.css
|
* commit 6dfaede781db0764781e0df6fff227f685039831
| Author: Алёна Горяйнова <alenagor2004@mail.ru>
| Date: Thu Feb 20 21:24:49 2025 +0300
|
| Added css stylesheet
|
* commit 6490608674720819c7f1f0bc670571c55f99dc1e
| Author: Алёна Горяйнова <alenagor2004@mail.ru>
| Date: Thu Feb 20 21:17:36 2025 +0300
|
|
[10]+ Остановлен git log --graph --all
[aagoryaynova@fedora hello]$ git checkout style
Уже на «style»
[aagoryaynova@fedora hello]$ git checkout master
Переключились на ветку «master»
[aagoryaynova@fedora hello]$ git merge style
```

Рис. 3.36: Поиск коммита перед конфликтом

Коммит «Added README» идет непосредственно перед коммитом конфликтующего интерактивного режима. Мы сбросим ветку master к коммиту «Added README»(рис. fig. 3.37).


```

*   commit 1077b37610f752e06ebb5c214d027e2bf84ee453 (HEAD -> style)
|   Merge: 74dc620 8dbd09b
|   Author: Алёна Горяйнова <alenagor2004@mail.ru>
|   Date: Thu Feb 20 21:40:11 2025 +0300
|
|       Merged master fixed conflict.
|
*   commit 8dbd09b302dc32480171d8fb99c98886c2f9b1d2 (master)
|   Author: Алёна Горяйнова <alenagor2004@mail.ru>
|   Date: Thu Feb 20 21:35:40 2025 +0300
|
|       Life is great
|
*   commit 74dc620ae681e3150ac0912c45bc9f42abe3a0a2
|   Author: Алёна Горяйнова <alenagor2004@mail.ru>
|   Date: Thu Feb 20 21:29:53 2025 +0300
|
|       Updated index.html
|
*   commit d581604553d22f90b0726650d9f3e4bdcf4c5c03
|   Author: Алёна Горяйнова <alenagor2004@mail.ru>
|   Date: Thu Feb 20 21:27:30 2025 +0300
|
|       Hello uses style.css
|
*   commit 6dfaede781db0764781e0df6fff227f685039831
|   Author: Алёна Горяйнова <alenagor2004@mail.ru>
|   Date: Thu Feb 20 21:24:49 2025 +0300
|
|       Added css stylesheet
|
*   commit 6490608674720819c7f1f0bc670571c55f99dc1e
|   Author: Алёна Горяйнова <alenagor2004@mail.ru>
|   Date: Thu Feb 20 21:17:36 2025 +0300
|
|       :
[10]+ Остановлен git log --graph --all
[aagoryaynova@fedora hello]$ git checkout style
Уже на «style»
[aagoryaynova@fedora hello]$ git checkout master
Переключились на ветку «master»
[aagoryaynova@fedora hello]$ git merge style

```

Рис. 3.37: Сброс ветки master

3.25 Перебазирование

Используем команду `rebase` вместо команды `merge`. Мы вернулись в точку до первого слияния и хотим перенести изменения из ветки `master` в нашу ветку `style`. На этот раз для переноса изменений из ветки `master` мы будем использовать команду `git rebase` вместо слияния(рис. fig. 3.38).

```

* commit 6490608674720819c7f1f0bc670571c55f99dc1e
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 21:17:36 2025 +0300
:
[10]+ Остановлен git log --graph --all
[aagoryayjnova@fedora hello]$ git checkout style
Уже на «style»
[aagoryayjnova@fedora hello]$ git checkout master
Переключились на ветку «master»
[aagoryayjnova@fedora hello]$ git merge style
Обновление 8dbd09b..1077b37
Fast-forward
 index.html | 4 ++++
 lib/hello.html | 3 ++-
 lib/style.css | 3 +++
 3 files changed, 9 insertions(+), 1 deletion(-)
 create mode 100644 lib/style.css
[aagoryayjnova@fedora hello]$ cd ..
[aagoryayjnova@fedora ~]$ pwd
/home/aagoryayjnova
[aagoryayjnova@fedora ~]$ ls

```

Рис. 3.38: Перебазирование

3.26 Слияние в ветку master

Вернемся в ветку master и сольем ветку style в неё с помощью команды git merge(рис. fig. 3.39).

```

lib/style.css | 3 +++
3 files changed, 9 insertions(+), 1 deletion(-)
create mode 100644 lib/style.css
[aagoryayjnova@fedora hello]$ cd ..
[aagoryayjnova@fedora ~]$ pwd
/home/aagoryayjnova
[aagoryayjnova@fedora ~]$ ls
---      hello      reports      Документы      Общедоступные
bin      lab07.sh~      Shakespeare  Загрузки      'Рабочий стол'
branching1.sh  lab14      work      Изображения      степик
cprog      presentation.md  Видео      Музыка      Шаблоны
[aagoryayjnova@fedora ~]$ git clone hello cloned_hello
Клонирование в «cloned_hello»...
готово.

```

Рис. 3.39: Слияние style в master

3.27 Клонирование репозитория

Перейдем в наш рабочий каталог и сделаем клон репозитория hello, затем создадим клон репозитория. Просмотрев его увидим список всех файлов на

верхнем уровне оригинального репозитория README.md, index.html и lib. Затем посмотрим историю репозитория и увидим список всех коммитов в новый репозиторий, и он совпадает с историей коммитов в оригинальном репозитории. Единствен в названиях веток(рис. fig. 3.40-[fig041?]).

```
[aagoryayjnova@fedora ~]$ git clone hello cloned_hello
Клонирование в «cloned_hello»...
готово.
[aagoryayjnova@fedora ~]$ ls
---      hello      Shakespeare  Изображения  Шаблоны
bin      lab07.sh~      work      Музыка
branching1.sh  lab14      Видео      Общедоступные
cloned_hello  presentation.md  Документы  'Рабочий стол'
cprog      reports      Загрузки  степик
[aagoryayjnova@fedora ~]$ cd cloned_hello
[aagoryayjnova@fedora cloned_hello]$ ls
index.html  lib
[aagoryayjnova@fedora cloned_hello]$ git remote
origin
[aagoryayjnova@fedora cloned_hello]$ git remote show origin
```

Рис. 3.40: Клонирование репозитория

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

Рис. 3.41: Клонирование репозитория

3.28 Что такое origin?

Клонированный репозиторий знает об имени по умолчанию удаленного репозитория. Посмотрим, подробную информацию об имени по умолчанию. Для того, чтобы увидеть все ветки используем опцию -a(рис. fig. 3.42).

```

[aagoryayjnova@fedora cloned_hello]$ cd ../hello
[aagoryayjnova@fedora hello]$ cat README.md
This is the Hello World example from the git tutorial.
[aagoryayjnova@fedora hello]$ git add README
fatal: спецификатор пути «README» не соответствует ни одному файлу
[aagoryayjnova@fedora hello]$ git add README.md
[aagoryayjnova@fedora hello]$ git commit -m "Changed README in original repo"
[master b917308] Changed README in original repo
1 file changed, 1 insertion(+)
create mode 100644 README.md
[aagoryayjnova@fedora hello]$ cd ../cloned_hello
[aagoryayjnova@fedora cloned_hello]$ git fetch
remote: Перечисление объектов: 4, готово.
remote: Подсчет объектов: 100% (4/4), готово.

```

Рис. 3.42: Просмотр имени по умолчанию удаленного репозитория

3.29 Удаленные ветки

Посмотрим на ветки, доступные в нашем клонированном репозитории. Можно увидеть, что в списке только ветка master.

3.30 Изменение оригинального репозитория

Перейдем в репозиторий hello. Внесем изменения в файл README.md. Затем добавим их в репозиторий (рис. fig. 3.43).

```

[aagoryayjnova@fedora cloned_hello]$ git fetch
remote: Перечисление объектов: 4, готово.
remote: Подсчет объектов: 100% (4/4), готово.
remote: Сжатие объектов: 100% (3/3), готово.
remote: Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно
        использовано пакетов 0
Распаковка объектов: 100% (3/3), 1.03 КиБ | 1.03 МиБ/с, готово.
Из /home/aagoryayjnova/hello
1077b37..b917308 master -> origin/master
[aagoryayjnova@fedora cloned_hello]$ git log --all
commit b917308d4cdc17227e9bc08f335bccf2e9239db3 (origin/master, origin/HEAD)
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 21:48:33 2025 +0300

    Changed README in original repo

commit 1077b37610f752e06ebb5c214d027e2bf84ee453 (HEAD -> master, origin/style)
Merge: 74dc620 8dbd09b
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 21:40:11 2025 +0300

    Merged master fixed conflict.

```

Рис. 3.43: Изменение оригинального репозитория

Перейдём в клон репозитория и используем команду `git fetch`, которая будет извлекать новые коммиты из удаленного репозитория, но не будет сливать их с наработками в локальных ветках(рис. fig. 3.44).

```
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 21:24:49 2025 +0300
:
[11]+ Остановлен git log --all
[aagoryayjnova@fedora cloned_hello]$ cat README
cat: README: Нет такого файла или каталога
[aagoryayjnova@fedora cloned_hello]$ git merge origin/master
Обновление 1077b37..b917308
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
[aagoryayjnova@fedora cloned_hello]$ cat README.md
This is the Hello World example from the git tutorial.
[aagoryayjnova@fedora cloned_hello]$ git pull
Уже актуально.
[aagoryayjnova@fedora cloned_hello]$ git branch --track style origin/style
branch 'style' set up to track 'origin/style'.
```

Рис. 3.44: Извлечение изменений

3.31 Слияние извлеченных изменений

Сольём внесённые изменения в главную ветку. Также можно было бы использовать команду `git pull`, которая является объединением `fetch` и `merge` в одну команду (рис. fig. 3.45).

```
This is the Hello World example from the git tutorial.
[aagoryayjnova@fedora cloned_hello]$ git pull
Уже актуально.
[aagoryayjnova@fedora cloned_hello]$ git branch --track style origin/style
branch 'style' set up to track 'origin/style'.
[aagoryayjnova@fedora cloned_hello]$ git branch -a
* master
  style
remotes/origin/HEAD -> origin/master
remotes/origin/master
remotes/origin/style
[aagoryayjnova@fedora cloned_hello]$ git log --max-count=2
commit b917308d4cdc17227e9bc08f335bccf2e9239db3 (HEAD -> master, origin/master, origin/HEAD)
Author: Алёна Горяйнова <alenagor2004@mail.ru>
Date: Thu Feb 20 21:48:33 2025 +0300

    Changed README in original repo
```

Рис. 3.45: Слияние извлеченных изменений

3.32 Добавление ветки наблюдения

Добавим локальную ветку, которая отслеживает удаленную ветку, теперь мы можем видеть ветку style в списке веток и логе(рис. fig. 3.46).

```
[aagoryayjnova@fedora cloned_hello]$ cd ..
[aagoryayjnova@fedora ~]$ git clone --bare hello hello.git
Клонирование в голый репозиторий «hello.git»...
готово.
[aagoryayjnova@fedora ~]$ ls hello.git
branches  config  description  HEAD  hooks  info  objects  packed-refs  refs
[aagoryayjnova@fedora ~]$ cd hello
[aagoryayjnova@fedora hello]$ git remote add shared ../hello.git
```

Рис. 3.46: Добавление ветки наблюдения

3.33 Создание чистого репозитория

Как правило, репозитории, оканчивающиеся на .git являются чистыми репозиториями. Создадим такой в рабочем каталоге. Затем добавим репозиторий hello.git к нашему оригинальному репозиторию(рис. fig. 3.47).

```
[aagoryayjnova@fedora hello]$ nano README.md
[aagoryayjnova@fedora hello]$ git checkout master
M      README.md
Уже на «master»
[aagoryayjnova@fedora hello]$ git add README
fatal: спецификатор пути «README» не соответствует ни одному файлу
[aagoryayjnova@fedora hello]$ git add README.md
[aagoryayjnova@fedora hello]$ git commit -m "Added shared comment to readme"
[master 5825511] Added shared comment to readme
1 file changed, 1 insertion(+)
[aagoryayjnova@fedora hello]$ git push shared master
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 1.08 КиБ | 1.08 МИБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To ../hello.git
```

Рис. 3.47: Создание чистого репозитория

3.34 Отправка и извлечение изменений

Так как чистые репозитории, как правило, расшариваются на каком-нибудь сетевом сервере, нам необходимо отправить наши изменения в другие репо-

зитории. Начнем с создания изменения для отправки. Отредактируем файл README.md и сделаем коммит, затем отправим изменения в общий репозиторий. Затем извлечем изменения из общего репозитория(рис. fig. 3.48, fig. 3.49).

```

Всего 3 (изменения 0), повторно использовано 0 (изменения 0), повторно использо
ано пакетов 0
To ../hello.git
b917308..5825511 master -> master
[aagoryayjnova@fedora hello]$ cd ../cloned_hello
[aagoryayjnova@fedora cloned_hello]$ git remote add shared ../hello.git
[aagoryayjnova@fedora cloned_hello]$ git branch --track shared master
branch 'shared' set up to track 'master'.
[aagoryayjnova@fedora cloned_hello]$ git pull shared master
remote: Перечисление объектов: 5, готово.
remote: Подсчет объектов: 100% (5/5), готово.
remote: Сжатие объектов: 100% (3/3), готово.
remote: Всего 3 (изменения 0), повторно использовано 0 (изменения 0), повторно
использовано пакетов 0
Распаковка объектов: 100% (3/3), 1.06 КиБ | 543.00 КиБ/с, готово.
Из ../hello
* branch          master      -> FETCH_HEAD
* [новая ветка]    master      -> shared/master
Обновление b917308..5825511
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
[aagoryayjnova@fedora cloned_hello]$ cat README.md
This is the Hello World example from the git tutorial.
(Changed in the original and pushed to shared)
[aagoryayjnova@fedora cloned_hello]$

```

Рис. 3.48: Отправка изменений

```

Всего 3 (изменения 0), повторно использовано 0 (изменения 0), повторно использо
ано пакетов 0
To ../hello.git
b917308..5825511 master -> master
[aagoryayjnova@fedora hello]$ cd ../cloned_hello
[aagoryayjnova@fedora cloned_hello]$ git remote add shared ../hello.git
[aagoryayjnova@fedora cloned_hello]$ git branch --track shared master
branch 'shared' set up to track 'master'.
[aagoryayjnova@fedora cloned_hello]$ git pull shared master
remote: Перечисление объектов: 5, готово.
remote: Подсчет объектов: 100% (5/5), готово.
remote: Сжатие объектов: 100% (3/3), готово.
remote: Всего 3 (изменения 0), повторно использовано 0 (изменения 0), повторно
использовано пакетов 0
Распаковка объектов: 100% (3/3), 1.06 КиБ | 543.00 КиБ/с, готово.
Из ../hello
* branch          master      -> FETCH_HEAD
* [новая ветка]    master      -> shared/master
Обновление b917308..5825511
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
[aagoryayjnova@fedora cloned_hello]$ cat README.md
This is the Hello World example from the git tutorial.
(Changed in the original and pushed to shared)
[aagoryayjnova@fedora cloned_hello]$

```

Рис. 3.49: Извлечение изменений

4 Выводы

В процессе выполнения данной лабораторной работы я приобрела практические навыки работы с Git. # Список литературы{unnumbered}