

Лабораторная работа N1

Работа с git

Туем Гислен

Содержание

1	Цель работы	8
2	Теоретическое введение	9
3	Выполнение лабораторной работы	10
3.1	Подготовка	10
3.1.1	Утановка имени и электронной почты	10
3.1.2	Параметры установки окончаний строк и Установка отображения unicode.(рис. fig23.1).	10
3.2	Создание проекта	11
3.2.1	Создайте страницу «Hello, World» (рис. 3.2).	11
3.2.2	Создание репозитория, добавление файла в репозиторий и Проверка состояние репозитория(рис. 3.3).	11
3.3	Внесение изменений	12
3.3.1	Измените страницу «Hello, World» и Проверьте состояние рабочего каталога.(рис. 3.4)	12
3.4	Индексация изменений	12
3.4.1	Коммит изменений	13
3.4.2	Добавьте стандартные теги страницы	14
3.4.3	История	17
3.4.4	Получение старых версий(рис. 3.16)	18
3.4.5	Создание тегов версий	20
3.4.6	Переключение по имени тега	21
3.4.7	Просмотр тегов с помощью команды tag	22
3.5	Отмена локальных изменений (до индексации)	22
3.5.1	Переключитесь на ветку master	22
3.5.2	Измените hello.html	23
3.5.3	Отмена изменений в рабочем каталоге	24
3.6	Отмена проиндексированных изменений (перед коммитом)	24
3.6.1	Измените файл и проиндексируйте изменения	24
3.6.2	Выполните сброс буферной зоны	25
3.6.3	Переключитесь на версию коммита	26
3.7	Отмена коммитов	26
3.7.1	Отмена коммитов	26
3.7.2	Измените файл и сделайте коммит	27
3.7.3	Сделайте коммит с новыми изменениями, отменяющими предыдущие	28

3.7.4	Проверьте лог	28
3.8	Удаление коммитов из ветки	29
3.8.1	Для начала отметьте эту ветку	29
3.8.2	Сброс коммитов к предшествующим коммиту Oops	30
3.9	Удаление тега oops	31
3.9.1	Удаление тега oops	31
3.10	Внесение изменений в коммиты	31
3.10.1	Измените страницу, а затем сделайте коммит	31
3.10.2	Необходим email	32
3.10.3	Измените предыдущий коммит	32
3.11	Перемещение файлов	33
3.11.1	Переместите файл hello.html в каталог lib	33
3.11.2	Второй способ перемещения файлов	33
3.11.3	Коммит в новый каталог(рис. 3.41)	33
3.12	Подробнее о структуре	34
3.12.1	Добавление index.html	34
3.13	Git внутри: Каталог .git	34
3.13.1	Каталог .git	34
3.13.2	Config File	35
3.13.3	Ветки и теги, Файл HEAD	35
3.14	Работа непосредственно с объектами git	36
3.14.1	Поиск последнего коммита (рис. 3.48)	36
3.14.2	Вывод последнего коммита с помощью SHA1 хэша	36
3.14.3	Поиск дерева	37
3.14.4	Вывод каталога lib, Вывод файла hello.html	37
3.15	Создание ветки, Добавьте файл стилей style.css, Измените основную страницу и Измените index.html	38
3.16	Навигация по веткам	40
3.16.1	Переключение на ветку master	41
3.16.2	Вернемся к ветке style	41
3.17	Изменения в ветке master	42
3.17.1	Создайте файл README в ветке master(рис. 3.62)	42
3.17.2	Сделайте коммит изменений README.md в ветку master.(рис. 3.63)	42
3.18	Слияние веток	43
3.19	Создание конфликта	43
3.19.1	Вернитесь в master и создайте конфликт	43
3.20	Разрешение конфликтов	44
3.20.1	Слияние master с веткой style	44
3.20.2	Решение конфликта	45
3.20.3	Сделайте коммит решения конфликта (рис. 3.67)	45
3.21	Клонирование репозитория	46
3.21.1	Перейдите в рабочий каталог, Создайте клон репозитория hello и давайте взглянем на клонированный репозиторий.	46

3.22 Что такое origin?	47
3.23 Извлечение общих изменений(рис. 3.73)	47
4 Выводы	49

Список иллюстраций

3.1	fig1	10
3.2	fig3	11
3.3	fig4	12
3.4	fig1	12
3.5	fig5	13
3.6	fig7	13
3.7	fig6	14
3.8	fig8	14
3.9	fig9	15
3.10	fig10	15
3.11	fig11	16
3.12	fig12	16
3.13	fig13	16
3.14	fig15	17
3.15	fig19	18
3.16	fig21	19
3.17	fig22	20
3.18	fig24	21
3.19	fig26	21
3.20	fig27	22
3.21	fig28	22
3.22	fig29	23
3.23	fig30	23
3.24	fig31	24
3.25	fig32	25
3.26	fig33	25
3.27	fig34	26
3.28	fig35	26
3.29	fig36	27
3.30	fig37	27
3.31	fig39	28
3.32	fig40	29
3.33	fig42	30
3.34	fig43	30
3.35	fig45	31
3.36	fig46	31
3.37	fig47	32

3.38 fig48	32
3.39 fig50	32
3.40 fig52	33
3.41 fig53	33
3.42 fig54	34
3.43 fig57	34
3.44 fig58	34
3.45 fig59	35
3.46 fig60	35
3.47 fig61	36
3.48 fig62	36
3.49 fig63	36
3.50 fig64	37
3.51 fig66	37
3.52 fig68	38
3.53 fig69	38
3.54 fig70	39
3.55 fig72	39
3.56 fig73	39
3.57 fig74	39
3.58 fig75	40
3.59 fig76	40
3.60 fig77	41
3.61 fig78	42
3.62 fig79	42
3.63 fig80	43
3.64 fig82	43
3.65 fig84	44
3.66 fig86	44
3.67 fig86	45
3.68 fig96	46
3.69 fig97	46
3.70 fig98	46
3.71 fig99	47
3.72 fig101	47
3.73 fig119	48

Список таблиц

1 Цель работы

Приобретить практические навыки работы с Git.

2 Теоретическое введение

Git-это мощный инструмент, созданный 7 апреля 2005 года для управления ядром linux.

3 Выполнение лабораторной работы

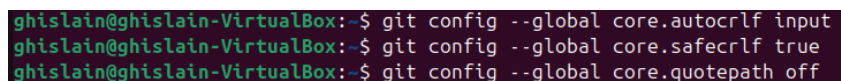
3.1 Подготовка

3.1.1 Установка имени и электронной почты

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

Если git уже установлен, можете переходить к разделу окончания строк.

3.1.2 Параметры установки окончаний строк и Установка отображения unicode.(рис. fig23.1).

A screenshot of a terminal window with a dark background and light-colored text. It shows three lines of commands being executed in a shell. The prompt is 'ghislain@ghislain-VirtualBox:~\$'. The commands are: 'git config --global core.autocrlf input', 'git config --global core.safecrlf true', and 'git config --global core.quotepath off'.

```
ghislain@ghislain-VirtualBox:~$ git config --global core.autocrlf input  
ghislain@ghislain-VirtualBox:~$ git config --global core.safecrlf true  
ghislain@ghislain-VirtualBox:~$ git config --global core.quotepath off
```

Рис. 3.1: fig1

Настройка core.autocrlf с параметрами true и input делает все переводы строк текстовых файлов в главном репозитории одинаковыми.

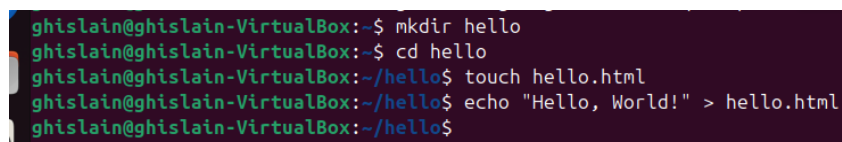
core.autocrlf true - git автоматически конвертирует CRLF->LF при коммите и обратно LF->CRLF при выгрузке кода из репозитория на файловую систему (используют в Windows). core.autocrlf input - конвертация CRLF в LF только при коммитах (используют в Mac/Linux).

```
git config --global core.autocrlf input
git config --global core.safecrlf true
git config --global core.quotepath off
```

3.2 Создание проекта

3.2.1 Создайте страницу «Hello, World» (рис. 3.2).

```
mkdir hello
cd hello
touch hello.html
echo "Hello, World!" > hello.html
```



```
ghislain@ghislain-VirtualBox:~$ mkdir hello
ghislain@ghislain-VirtualBox:~$ cd hello
ghislain@ghislain-VirtualBox:~/hello$ touch hello.html
ghislain@ghislain-VirtualBox:~/hello$ echo "Hello, World!" > hello.html
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.2: fig3

3.2.2 Создание репозитория, добавление файла в репозиторий и Проверка состояние репозитория(рис. 3.3).

```
git init
git add hello.html
git commit -m "Initial Commit"
git status
```

```
ghislain@ghislain-VirtualBox:~/hello$ git init
Initialized empty Git repository in /home/ghislain/hello/.git/
ghislain@ghislain-VirtualBox:~/hello$ git add hello.html
ghislain@ghislain-VirtualBox:~/hello$ git commit -m "Initial Commit"
[master (root-commit) 985f8dd] Initial Commit
1 file changed, 1 insertion(+)
create mode 100644 hello.html
ghislain@ghislain-VirtualBox:~/hello$ git status
On branch master
nothing to commit, working tree clean
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.3: fig4

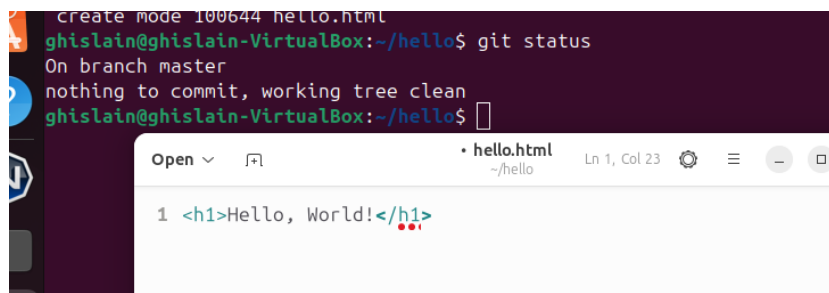
3.3 Внесение изменений

3.3.1 Измените страницу «Hello, World» и Проверьте состояние рабочего каталога.(рис. 3.4)

Добавим кое-какие HTML-теги к нашему приветствию. Измените содержимое файла hello.html на:

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

```
git status
```



```
create mode 100644 hello.html
ghislain@ghislain-VirtualBox:~/hello$ git status
On branch master
nothing to commit, working tree clean
ghislain@ghislain-VirtualBox:~/hello$
```

hello.html
Ln 1, Col 23

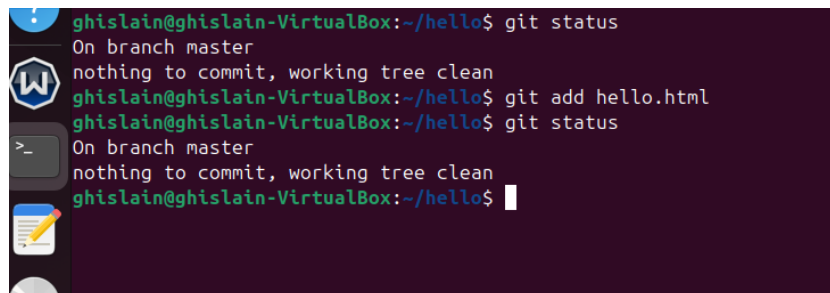
```
1 <h1>Hello, World!</h1>
```

Рис. 3.4: fig1

3.4 Индексация изменений

Теперь выполните команду git, чтобы проиндексировать изменения. Проверьте состояние.(рис. 3.5).

```
git add hello.html
git status
```

A screenshot of a terminal window with a dark purple background. The prompt is 'ghislain@ghislain-VirtualBox:~/hello\$'. The first command is 'git status', which outputs 'On branch master' and 'nothing to commit, working tree clean'. The second command is 'git add hello.html'. The third command is 'git status', which again outputs 'On branch master' and 'nothing to commit, working tree clean'.

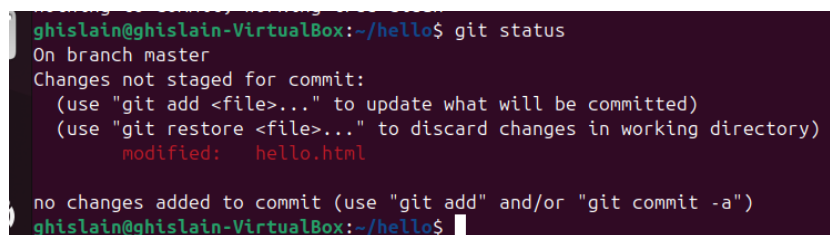
```
ghislain@ghislain-VirtualBox:~/hello$ git status
On branch master
nothing to commit, working tree clean
ghislain@ghislain-VirtualBox:~/hello$ git add hello.html
ghislain@ghislain-VirtualBox:~/hello$ git status
On branch master
nothing to commit, working tree clean
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.5: fig5

3.4.1 Коммит изменений

Сделайте коммит и проверьте состояние.(рис. 3.6).

```
git commit
```

A screenshot of a terminal window with a dark purple background. The prompt is 'ghislain@ghislain-VirtualBox:~/hello\$'. The command is 'git status'. The output shows 'On branch master' and 'Changes not staged for commit:'. It lists instructions for using 'git add' and 'git restore', and shows 'modified: hello.html'. It concludes with 'no changes added to commit (use "git add" and/or "git commit -a")'.

```
ghislain@ghislain-VirtualBox:~/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")
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.6: fig7

В первой строке введите комментарий: «Added h1 tag». Сохраните файл и выйдите из редактора.(рис. 3.7).

```
git status
```



Рис. 3.7: fig6

3.4.2 Добавьте стандартные теги страницы

Измените страницу «Hello, World», чтобы она содержала стандартные теги.(рис. 3.8)

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

A screenshot of a code editor window showing the same file 'hello.html'. The content now includes the full HTML structure: '1 Added h1 tag', '2 <html>', '3 <body>', '4 <h1>Hello, World!</h1>', '5 </body>', and '6 </html>'. Red dots are visible under the 'h1' in line 1 and under the closing tag '</h1>' in line 4.

Рис. 3.8: fig8

Теперь добавьте это изменение в индекс git.

```
git add hello.html
```

Теперь добавьте заголовки HTML (секцию

) к странице «Hello, World».(рис. 3.9)

```
1 Added h1 tag
2 <html>
3   <head>
4   </head>
5   <body>
6     <h1>Hello, World!</h1>
7   </body>
8 </html>
```

Рис. 3.9: fig9

Проверьте текущий статус (рис. 3.9):

```
git status
```

```
ghislain@ghislain-VirtualBox:~/hello$ git add hello.html
ghislain@ghislain-VirtualBox:~/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

ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.10: fig10

Произведите коммит проиндексированного изменения (значение по умолчанию), а затем еще раз проверьте состояние.(рис. 3.11)

```
git commit -m "Added standard HTML page tags"
git status
```

```
ghislain@ghislain-VirtualBox:~/hello$ git commit -m "Added standard HTML page tags"
[master e8cd743] Added standard HTML page tags
1 file changed, 6 insertions(+), 1 deletion(-)
ghislain@ghislain-VirtualBox:~/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")
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.11: fig11

Теперь добавьте второе изменение в индекс, а затем проверьте состояние с помощью команды `git status`. (рис. 3.12)

```
git add .
git status
```

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

ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.12: fig12

Сделайте коммит второго изменения (рис. 3.13)

```
git commit -m "Added HTML header"
```

```
ghislain@ghislain-VirtualBox:~/hello$ git commit -m "Added HTML header"
[master 19ebbc1] Added HTML header
1 file changed, 2 insertions(+)
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.13: fig13

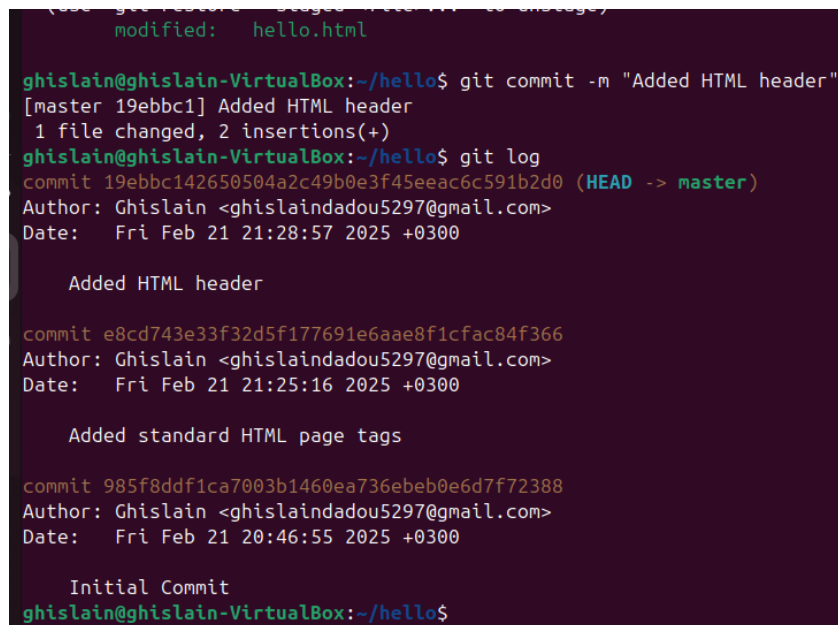
3.4.3 История

Получим список произведенных изменений(рис. 3.14):

```
git log
```

Есть много вариантов отображения лога.

```
git log --pretty=oneline --max-count=2
git log --pretty=oneline --since='5 minutes ago'
git log --pretty=oneline --until='5 minutes ago'
git log --pretty=oneline --author=<your name>
git log --pretty=oneline --all
```

A screenshot of a terminal window with a dark purple background. The terminal shows the output of a 'git log' command. It displays three commit entries in a compact format. Each entry includes a commit hash, a short description, the author's name and email, and the date. The first commit is the most recent, followed by the second, and then the initial commit at the bottom. The text is color-coded: commit hashes are green, descriptions are white, author information is yellow, and dates are white. The prompt 'ghislain@ghislain-VirtualBox:~/hello\$' is visible at the bottom of the terminal window.

```
(use git restore --staged <file> to unstage)
modified:   hello.html

ghislain@ghislain-VirtualBox:~/hello$ git commit -m "Added HTML header"
[master 19ebbc1] Added HTML header
1 file changed, 2 insertions(+)
ghislain@ghislain-VirtualBox:~/hello$ git log
commit 19ebbc142650504a2c49b0e3f45eeac6c591b2d0 (HEAD -> master)
Author: Ghislain <ghislaindadou5297@gmail.com>
Date:   Fri Feb 21 21:28:57 2025 +0300

    Added HTML header

commit e8cd743e33f32d5f177691e6aae8f1cfac84f366
Author: Ghislain <ghislaindadou5297@gmail.com>
Date:   Fri Feb 21 21:25:16 2025 +0300

    Added standard HTML page tags

commit 985f8ddf1ca7003b1460ea736eb0e6d7f72388
Author: Ghislain <ghislaindadou5297@gmail.com>
Date:   Fri Feb 21 20:46:55 2025 +0300

    Initial Commit
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.14: fig15

Справочная страница(рис. 3.15):

```
man git-log
```

```
GIT-LOG(1)                               Git Manual                               GIT-LOG(1)

NAME
  git-log - Show commit logs

SYNOPSIS
  git log [<options>] [<revision-range>] [--] <path>...

DESCRIPTION
  Shows the commit logs.

  List commits that are reachable by following the parent links
  from the given commit(s), but exclude commits that are
  reachable from the one(s) given with a ^ in front of them. The
  output is given in reverse chronological order by default.

  You can think of this as a set operation. Commits reachable
  from any of the commits given on the command line form a set,
  and then commits reachable from any of the ones given with ^
  in front are subtracted from that set. The remaining commits
  are what comes out in the command's output. Various other
  options and paths parameters can be used to further limit the
  result.

  Thus, the following command:

      $ git log foo bar ^baz

  means "list all the commits which are reachable from foo or
  bar, but not from baz".

  A special notation "<commit1>..<commit2>" can be used as a
  short-hand for "^<commit1> <commit2>". For example, either of
  the following may be used interchangeably:

      $ git log origin..HEAD

Manual page git-log(1) line 1 (press h for help or q to quit)
```

Рис. 3.15: fig19

3.4.4 Получение старых версий(рис. 3.16)

```
git log
git checkout <hash>
cat hello.htm
```

```
ghislain@ghislain-VirtualBox:~/hello$ git checkout 985f8dd
Note: switching to '985f8dd'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to
false

HEAD is now at 985f8dd Initial Commit
ghislain@ghislain-VirtualBox:~/hello$ cat hello.html
Hello, World!
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.16: fig21

Вернитесь к последней версии в ветке master(рис. 3.17)

```
git checkout master
```

```
cat hello.html
```

```

git switch -c <new-branch-name>

Or undo this operation with:

git switch -

Turn off this advice by setting config variable advice.detachedHead to
false

HEAD is now at 985f8dd Initial Commit
ghislain@ghislain-VirtualBox:~/hello$ cat hello.html
Hello, World!
ghislain@ghislain-VirtualBox:~/hello$ git checkout master
Previous HEAD position was 985f8dd Initial Commit
Switched to branch 'master'
ghislain@ghislain-VirtualBox:~/hello$ cat hello.html
Added h1 tag
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
ghislain@ghislain-VirtualBox:~/hello$

```

Рис. 3.17: fig22

3.4.5 Создание тегов версий

```
git tag v1
```

Теперь текущая версия страницы называется v1. Теги для предыдущих версий
Давайте создадим тег для версии, которая идет перед текущей версией и назовем
его v1-beta и сделаем ее версией v1-beta.(рис. 3.18)

```

git checkout v1^
cat hello.html
git tag v1-beta

```

```

ghislain@ghislain-VirtualBox:~/hello$ git tag v1
ghislain@ghislain-VirtualBox:~/hello$ git checkout v1^
Note: switching to 'v1^'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to
false

HEAD is now at e8cd743 Added standard HTML page tags
ghislain@ghislain-VirtualBox:~/hello$ cat hello.html
Added h1 tag
<html>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
ghislain@ghislain-VirtualBox:~/hello$ git tag v1-beta
ghislain@ghislain-VirtualBox:~/hello$

```

Рис. 3.18: fig24

3.4.6 Переключение по имени тега

Теперь попробуйте попереключаться между двумя отмеченными версиями.(рис. 3.19)

```
git checkout v1
```

```
git checkout v1-beta
```

```

ghislain@ghislain-VirtualBox:~/hello$ git tag v1-beta
ghislain@ghislain-VirtualBox:~/hello$ git checkout v1
Previous HEAD position was e8cd743 Added standard HTML page tags
HEAD is now at 19ebbc1 Added HTML header
ghislain@ghislain-VirtualBox:~/hello$ git checkout v1-beta
Previous HEAD position was 19ebbc1 Added HTML header
HEAD is now at e8cd743 Added standard HTML page tags
ghislain@ghislain-VirtualBox:~/hello$

```

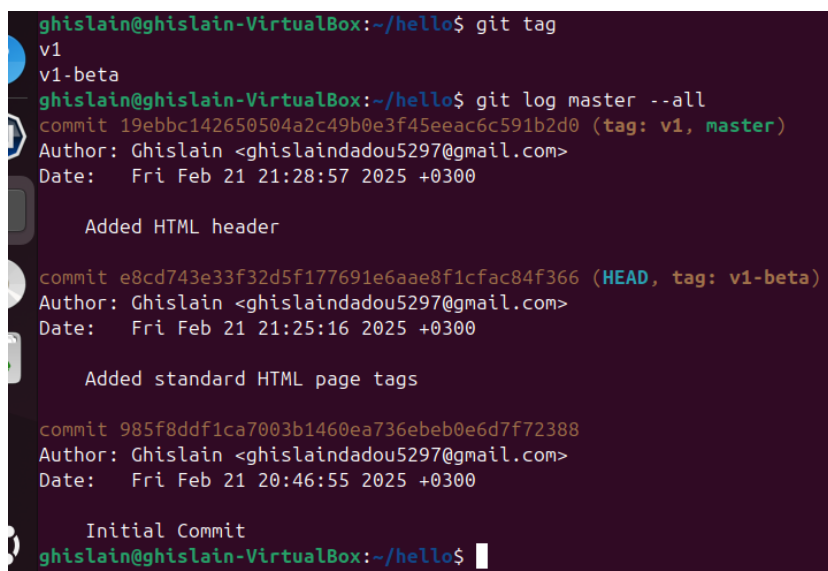
Рис. 3.19: fig26

3.4.7 Просмотр тегов с помощью команды tag

Вы можете увидеть, какие теги доступны, используя команду `git tag` и также можете посмотреть теги в логе. (рис. 3.20)

```
git tag
```

```
git log master --all
```



```
ghislain@ghislain-VirtualBox:~/hello$ git tag
v1
v1-beta
ghislain@ghislain-VirtualBox:~/hello$ git log master --all
commit 19ebbc142650504a2c49b0e3f45eeac6c591b2d0 (tag: v1, master)
Author: Ghislain <ghislaindadou5297@gmail.com>
Date:   Fri Feb 21 21:28:57 2025 +0300

    Added HTML header

commit e8cd743e33f32d5f177691e6aae8f1cfac84f366 (HEAD, tag: v1-beta)
Author: Ghislain <ghislaindadou5297@gmail.com>
Date:   Fri Feb 21 21:25:16 2025 +0300

    Added standard HTML page tags

commit 985f8dddf1ca7003b1460ea736eb0e6d7f72388
Author: Ghislain <ghislaindadou5297@gmail.com>
Date:   Fri Feb 21 20:46:55 2025 +0300

    Initial Commit
ghislain@ghislain-VirtualBox:~/hello$
```

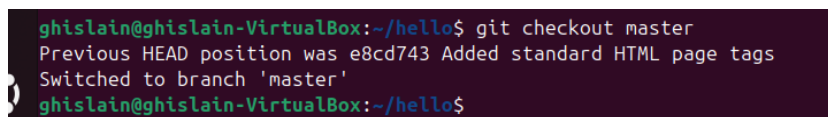
Рис. 3.20: fig27

3.5 Отмена локальных изменений (до индексации)

3.5.1 Переключитесь на ветку master

Убедитесь, что вы находитесь на последнем коммите ветки `master`, прежде чем продолжить работу. (рис. 3.21)

```
git checkout master
```



```
ghislain@ghislain-VirtualBox:~/hello$ git checkout master
Previous HEAD position was e8cd743 Added standard HTML page tags
Switched to branch 'master'
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.21: fig28

3.5.2 Измените hello.html

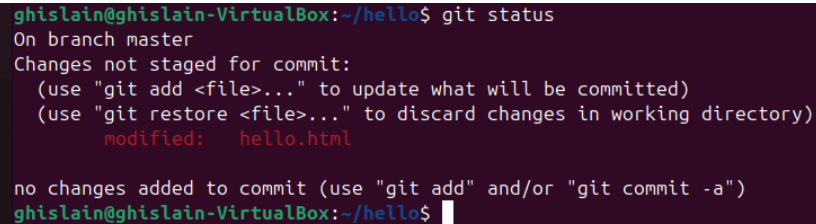
Внесите изменение в файл hello.html в виде нежелательного комментария.(рис. 3.22) и проверьте состояние.(рис. 3.23)

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



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

Рис. 3.22: fig29



```
ghislain@ghislain-VirtualBox:~/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")
ghislain@ghislain-VirtualBox:~/hello$
```

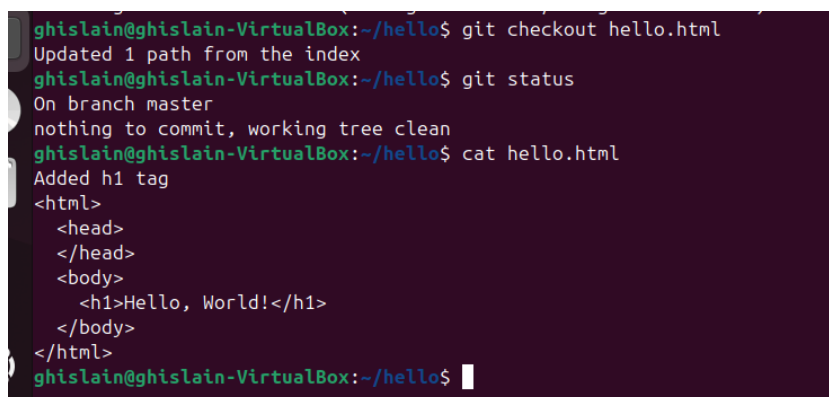
Рис. 3.23: fig30

Мы видим, что файл hello.html был изменен, но еще не проиндексирован.

3.5.3 Отмена изменений в рабочем каталоге

Используйте команду `git checkout` для переключения версии файла `hello.html` в репозитории.(рис. 3.24)

```
git checkout hello.html
git status
cat hello.html
```



```
ghislain@ghislain-VirtualBox:~/hello$ git checkout hello.html
Updated 1 path from the index
ghislain@ghislain-VirtualBox:~/hello$ git status
On branch master
nothing to commit, working tree clean
ghislain@ghislain-VirtualBox:~/hello$ cat hello.html
Added h1 tag
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.24: fig31

3.6 Отмена проиндексированных изменений (перед коммитом)

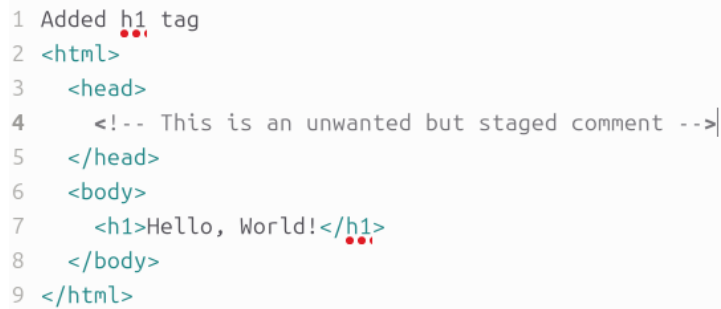
3.6.1 Измените файл и проиндексируйте изменения

Внесите изменение в файл `hello.html` в виде нежелательного комментария(рис. 3.25).

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



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

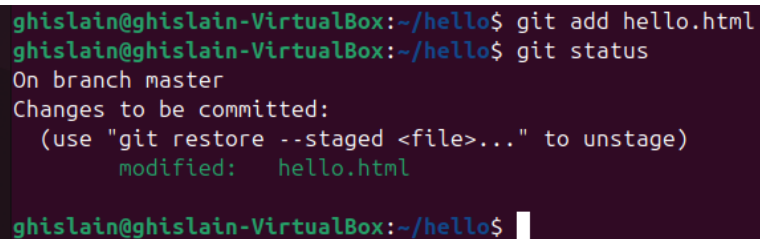


```
1 Added h1 tag
2 <html>
3   <head>
4     <!-- This is an unwanted but staged comment -->
5   </head>
6   <body>
7     <h1>Hello, World!</h1>
8   </body>
9 </html>
```

Рис. 3.25: fig32

Проиндексируйте это изменение и проверьте состояние(рис. 3.26).

```
git add hello.html
git status
```



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

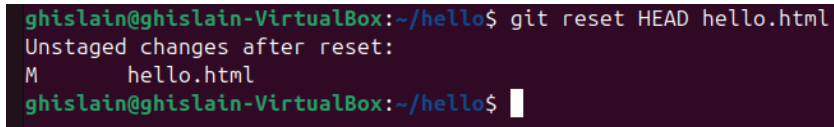
Рис. 3.26: fig33

3.6.2 Выполните сброс буферной зоны

К счастью, вывод состояния показывает нам именно то, что мы должны сделать для отмены индексации изменения.(рис. 3.27).

```
git reset HEAD hello.html
```

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. Мы можем использовать команду `git checkout`, чтобы удалить нежелательные изменения в рабочем каталоге.



```
ghislain@ghislain-VirtualBox:~/hello$ git reset HEAD hello.html
Unstaged changes after reset:
M   hello.html
ghislain@ghislain-VirtualBox:~/hello$
```

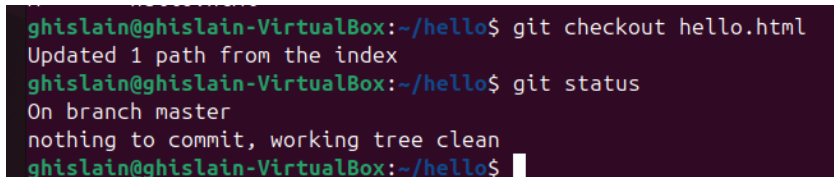
Рис. 3.27: fig34

3.6.3 Переключитесь на версию коммита

```
git checkout hello.html
```

```
git status
```

Наш рабочий каталог опять чист(рис. 3.28).



```
ghislain@ghislain-VirtualBox:~/hello$ git checkout hello.html
Updated 1 path from the index
ghislain@ghislain-VirtualBox:~/hello$ git status
On branch master
nothing to commit, working tree clean
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.28: fig35

3.7 Отмена коммитов

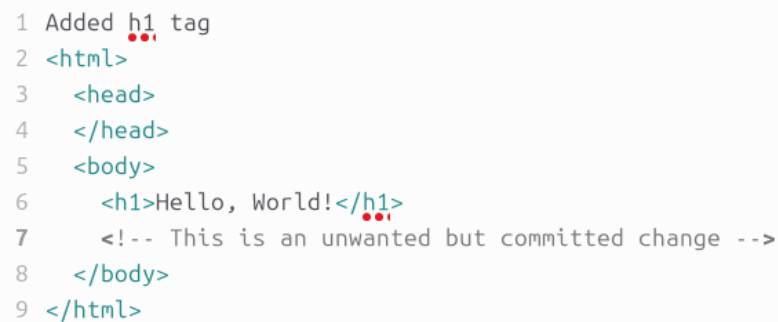
3.7.1 Отмена коммитов

Иногда вы понимаете, что новые коммиты являются неверными, и хотите их отменить. Есть несколько способов решения этого вопроса, здесь мы будем использовать самый безопасный. Мы отменим коммит путем создания нового коммита, отменяющего нежелательные изменения.

3.7.2 Измените файл и сделайте коммит

Измените файл hello.html на следующий(рис. 3.29).

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



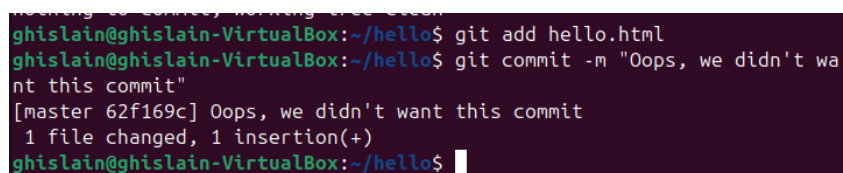
```
1 Added h1 tag
2 <html>
3   <head>
4   </head>
5   <body>
6     <h1>Hello, World!</h1>
7     <!-- This is an unwanted but committed change -->
8   </body>
9 </html>
```

Рис. 3.29: fig36

Выполните(рис. 3.30).:

```
git add hello.html
```

```
git commit -m "Oops, we didn't want this commit"
```



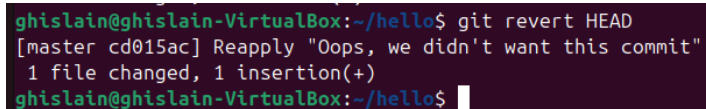
```
ghislain@ghislain-VirtualBox:~/hello$ git add hello.html
ghislain@ghislain-VirtualBox:~/hello$ git commit -m "Oops, we didn't want this commit"
[master 62f169c] Oops, we didn't want this commit
1 file changed, 1 insertion(+)
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.30: fig37

3.7.3 Сделайте коммит с новыми изменениями, отменяющими предыдущие

Чтобы отменить коммит, нам необходимо сделать коммит, который удаляет изменения, сохраненные нежелательным коммитом(рис. 3.31).

```
git revert HEAD
```



```
ghislain@ghislain-VirtualBox:~/hello$ git revert HEAD
[master cd015ac] Reapply "Oops, we didn't want this commit"
1 file changed, 1 insertion(+)
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.31: fig39

3.7.4 Проверьте лог

```
git log
```

Эта техника будет работать с любым коммитом(рис. 3.32).

```

ghislain@ghislain-VirtualBox:~/hello$ git log
commit cd015ac9d9ae7ed917184b8b2ab0e19779f84ca4 (HEAD -> master)
Author: Ghislain <ghislaindadou5297@gmail.com>
Date:   Fri Feb 21 22:42:41 2025 +0300

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

    This reverts commit c1cb7f9438631b6efb7c321c4375eb8912f99b1b.
    ^

commit c1cb7f9438631b6efb7c321c4375eb8912f99b1b
Author: Ghislain <ghislaindadou5297@gmail.com>
Date:   Fri Feb 21 22:31:01 2025 +0300

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

    This reverts commit 62f169cc45aacd1553413920e54246c00226ea86.

commit 62f169cc45aacd1553413920e54246c00226ea86
Author: Ghislain <ghislaindadou5297@gmail.com>
Date:   Fri Feb 21 22:23:18 2025 +0300

    Oops, we didn't want this commit

commit 19ebbc142650504a2c49b0e3f45eeac6c591b2d0 (tag: v1)
Author: Ghislain <ghislaindadou5297@gmail.com>
Date:   Fri Feb 21 21:28:57 2025 +0300

    Added HTML header

commit e8cd743e33f32d5f177691e6aae8f1cfac84f366 (tag: v1-beta)
Author: Ghislain <ghislaindadou5297@gmail.com>
Date:   Fri Feb 21 21:25:16 2025 +0300

    Added standard HTML page tags

```

Рис. 3.32: fig40

3.8 Удаление коммиттов из ветки

3.8.1 Для начала отметьте эту ветку

Но прежде чем удалить коммиты, давайте отметим последний коммит тегом, чтобы потом можно было его найти(рис. 3.33).

```
git tag oops
```

```
Initial Commit
ghislain@ghislain-VirtualBox:~/hello$ git tag oops
ghislain@ghislain-VirtualBox:~/hello$ git reset --hard v1
HEAD is now at 19ebbc1 Added HTML header
ghislain@ghislain-VirtualBox:~/hello$ git log
```

Рис. 3.33: fig42

3.8.2 Сброс коммитов к предшествующим коммиту Oops

```
git reset --hard v1
git log
```

Мы видим, что ошибочные коммиты не исчезли. Они все еще находятся в репозитории. Просто они отсутствуют в ветке master. Если бы мы не отметили их тегами, они по-прежнему находились бы в репозитории, но не было бы никакой возможности ссылаться на них, кроме как при помощи их хэш имен. Коммиты, на которые нет ссылок, остаются в репозитории до тех пор, пока не будет запущен сборщик мусора(рис. 3.34).

```
Initial Commit
ghislain@ghislain-VirtualBox:~/hello$ git tag oops
ghislain@ghislain-VirtualBox:~/hello$ git reset --hard v1
HEAD is now at 19ebbc1 Added HTML header
ghislain@ghislain-VirtualBox:~/hello$ git log
commit 19ebbc142650504a2c49b0e3f45eeac6c591b2d0 (HEAD -> master, tag: v1)
Author: Ghislain <ghislaindadou5297@gmail.com>
Date:   Fri Feb 21 21:28:57 2025 +0300

    Added HTML header

commit e8cd743e33f32d5f177691e6aae8f1cfac84f366 (tag: v1-beta)
Author: Ghislain <ghislaindadou5297@gmail.com>
Date:   Fri Feb 21 21:25:16 2025 +0300

    Added standard HTML page tags

commit 985f8ddf1ca7003b1460ea736eb0e6d7f72388
Author: Ghislain <ghislaindadou5297@gmail.com>
Date:   Fri Feb 21 20:46:55 2025 +0300

    Initial Commit
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.34: fig43

3.9 Удаление тега оорс

3.9.1 Удаление тега оорс

Тег оорс свою функцию выполнил. Давайте удалим его и коммиты, на которые он ссылался, сборщиком мусора(рис. 3.35).

```
ghislain@ghislain-VirtualBox:~/hello$ git tag -d oops
Deleted tag 'oops' (was cd015ac)
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.35: fig45

3.10 Внесение изменений в коммиты

3.10.1 Измените страницу, а затем сделайте коммит

(рис. 3.36).



```
★ S 1 Added h1 tag
H F 2 <!-- Author: tyem gislen -->|
D C 3 <html>
D C 4 <head>
D C 5 </head>
M N 6 <body>
F 7 <h1>Hello, World!</h1>
F 8 </body>
F V 9 </html>
```

Рис. 3.36: fig46

Выполните:

```
git add hello.html
```

```
git commit -m "Add an author comment"
```

(рис. 3.37).

```

Initial commit
ghislain@ghislain-VirtualBox:~/hello$ git add hello.html
ghislain@ghislain-VirtualBox:~/hello$ git commit -m "Add an author comment"
[master 8ac6e48] Add an author comment
1 file changed, 1 insertion(+)
ghislain@ghislain-VirtualBox:~/hello$

```

Рис. 3.37: fig47

3.10.2 Необходим email

(рис. 3.38).

```

1 Added h1 tag
2 <!-- Author: tyem gislen (tyemg@gmail.com) -->
3 <html>
4   <head>
5   </head>
6   <body>
7     <h1>Hello, World!</h1>
8   </body>
9 </html>

```

Рис. 3.38: fig48

3.10.3 Измените предыдущий коммит

(рис. 3.39).

```
git add hello.html
```

```
git commit --amend -m "Add an author/email comment"
```

```

ghislain@ghislain-VirtualBox:~/hello$ git add hello.html
ghislain@ghislain-VirtualBox:~/hello$ git commit --amend -m "Add an author/email
comment"
[master e433326] Add an author/email comment
Date: Fri Feb 21 23:07:55 2025 +0300
1 file changed, 1 insertion(+)
ghislain@ghislain-VirtualBox:~/hello$

```

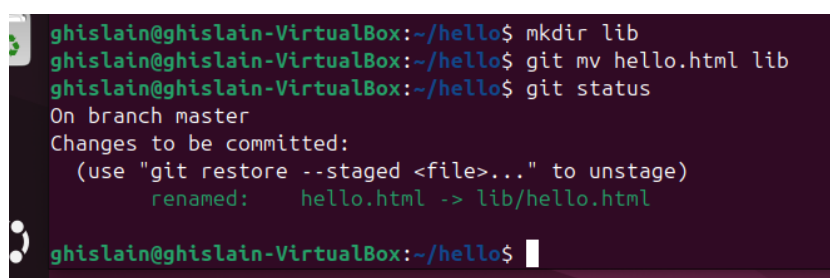
Рис. 3.39: fig50

3.11 Перемещение файлов

3.11.1 Переместите файл hello.html в каталог lib

Сейчас мы собираемся создать структуру нашего репозитория. Давайте перенесем страницу в каталог lib.(рис. 3.40).

```
mkdir lib
git mv hello.html lib
git status
```

A terminal window with a dark purple background. The prompt is 'ghislain@ghislain-VirtualBox:~/hello\$'. The user enters 'mkdir lib', then 'git mv hello.html lib', and finally 'git status'. The output of 'git status' shows 'On branch master' and 'Changes to be committed: (use "git restore --staged <file>..." to unstage) renamed: hello.html -> lib/hello.html'.

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

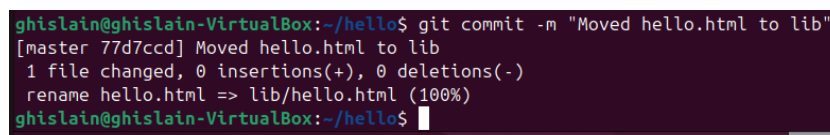
Рис. 3.40: fig52

3.11.2 Второй способ перемещения файлов

```
mkdir lib
mv hello.html lib
git add lib/hello.html
git rm hello.html
```

3.11.3 Коммит в новый каталог(рис. 3.41)

```
git commit -m "Moved hello.html to lib"
```

A terminal window with a dark purple background. The prompt is 'ghislain@ghislain-VirtualBox:~/hello\$'. The user enters 'git commit -m "Moved hello.html to lib"'. The output shows '[master 77d7ccd] Moved hello.html to lib', '1 file changed, 0 insertions(+), 0 deletions(-)', and 'rename hello.html => lib/hello.html (100%)'.

```
ghislain@ghislain-VirtualBox:~/hello$ git commit -m "Moved hello.html to lib"
[master 77d7ccd] Moved hello.html to lib
1 file changed, 0 insertions(+), 0 deletions(-)
rename hello.html => lib/hello.html (100%)
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.41: fig53

3.12 Подробнее о структуре

3.12.1 Добавление index.html

Добавим файл index.html в наш репозиторий и сделаем коммит (рис. 3.42)

```
ghislain@ghislain-VirtualBox:~/hello/lib$ touch index.html
ghislain@ghislain-VirtualBox:~/hello/lib$ git add index.html
ghislain@ghislain-VirtualBox:~/hello/lib$ git commit -m "Added index.html."
[master ba28612] Added index.html.
1 file changed, 5 insertions(+)
create mode 100644 lib/index.html
ghislain@ghislain-VirtualBox:~/hello/lib$
```

Рис. 3.42: fig54

3.13 Git внутри: Каталог .git

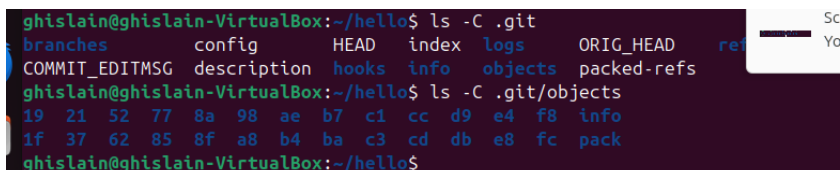
3.13.1 Каталог .git

Выполните: (рис. 3.43), (рис. 3.44)

```
ls -C .git
```

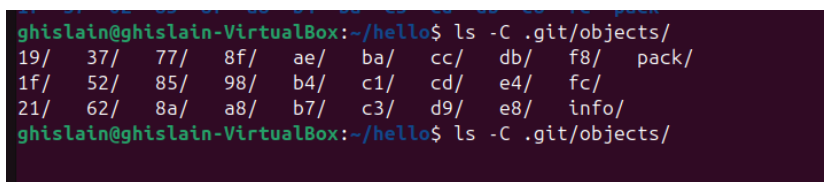
```
ls -C .git/objects
```

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



```
ghislain@ghislain-VirtualBox:~/hello$ ls -C .git
branches  config    HEAD     index    logs     ORIG_HEAD  refs
COMMIT_EDITMSG  description  hooks    info     objects  packed-refs
ghislain@ghislain-VirtualBox:~/hello$ ls -C .git/objects
19  21  52  77  8a  98  ae  b7  c1  cc  d9  e4  f8  info
1f  37  62  85  8f  a8  b4  ba  c3  cd  db  e8  fc  pack
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.43: fig57



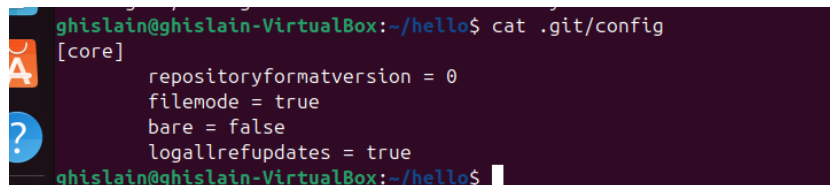
```
ghislain@ghislain-VirtualBox:~/hello$ ls -C .git/objects/
19/  37/  77/  8f/  ae/  ba/  cc/  db/  f8/  pack/
1f/  52/  85/  98/  b4/  c1/  cd/  e4/  fc/
21/  62/  8a/  a8/  b7/  c3/  d9/  e8/  info/
ghislain@ghislain-VirtualBox:~/hello$ ls -C .git/objects/
```

Рис. 3.44: fig58

3.13.2 Config File

Выполните:(рис. 3.45)

```
cat .git/config
```

A terminal window with a dark purple background. The prompt is 'ghislain@ghislain-VirtualBox:~/hello\$'. The command 'cat .git/config' has been executed, showing the following configuration for the [core] section: repositoryformatversion = 0, filemode = true, bare = false, logallrefupdates = true.

```
ghislain@ghislain-VirtualBox:~/hello$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.45: fig59

3.13.3 Ветки и теги, Файл HEAD

Выполните:(рис. 3.46), (рис. 3.47)

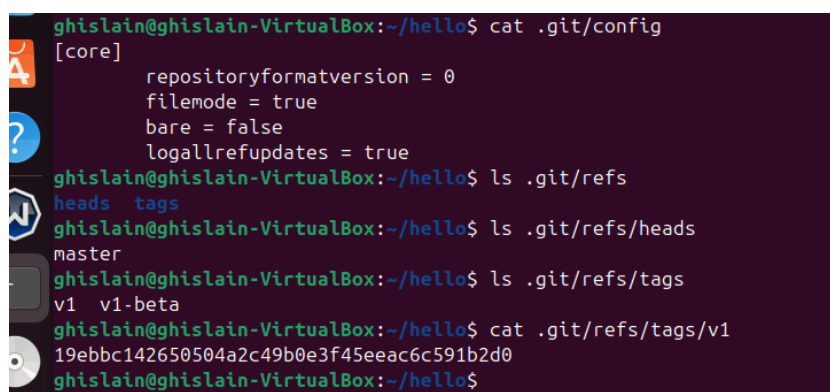
```
ls .git/refs
```

```
ls .git/refs/heads
```

```
ls .git/refs/tags
```

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

```
cat .git/HEAD
```

A terminal window with a dark purple background. The prompt is 'ghislain@ghislain-VirtualBox:~/hello\$'. The commands and their outputs are: 'cat .git/config' (same as previous), 'ls .git/refs' (output: heads tags), 'ls .git/refs/heads' (output: master), 'ls .git/refs/tags' (output: v1 v1-beta), and 'cat .git/refs/tags/v1' (output: 19ebbc142650504a2c49b0e3f45eeac6c591b2d0).

```
ghislain@ghislain-VirtualBox:~/hello$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
ghislain@ghislain-VirtualBox:~/hello$ ls .git/refs
heads tags
ghislain@ghislain-VirtualBox:~/hello$ ls .git/refs/heads
master
ghislain@ghislain-VirtualBox:~/hello$ ls .git/refs/tags
v1 v1-beta
ghislain@ghislain-VirtualBox:~/hello$ cat .git/refs/tags/v1
19ebbc142650504a2c49b0e3f45eeac6c591b2d0
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.46: fig60

```
ghislain@ghislain-VirtualBox:~/hello$ cat .git/HEAD
ref: refs/heads/master
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.47: fig61

3.14 Работа непосредственно с объектами git

3.14.1 Поиск последнего коммита (рис. 3.48)

`git log --max-count=1`

```
ghislain@ghislain-VirtualBox:~/hello$ git log --max-count=1
commit ba28612f64acc22899a5f444a60595355bfb84e8 (HEAD -> master)
Author: Ghislain <ghislaindadou5297@gmail.com>
Date:   Fri Feb 21 23:53:28 2025 +0300

    Added index.html.
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.48: fig62

3.14.2 Вывод последнего коммита с помощью SHA1 хэша

Выполните:(рис. 3.49)

`git cat-file -t <hash>`

`git cat-file -p <hash>`

```
ghislain@ghislain-VirtualBox:~/hello$ git log --max-count=1
commit ba28612f64acc22899a5f444a60595355bfb84e8 (HEAD -> master)
Author: Ghislain <ghislaindadou5297@gmail.com>
Date:   Fri Feb 21 23:53:28 2025 +0300

    Added index.html.
ghislain@ghislain-VirtualBox:~/hello$ git cat-file -t ba28612
commit
ghislain@ghislain-VirtualBox:~/hello$ git cat-file -p ba28612
tree ccc73963726c5b183edf37f1412dcb9ab45bfefa
parent 77d7ccd608a8a2ab35b60cb3d8ae2f5c385c2db7
author Ghislain <ghislaindadou5297@gmail.com> 1740171208 +0300
committer Ghislain <ghislaindadou5297@gmail.com> 1740171208 +0300

    Added index.html
```

Рис. 3.49: fig63

3.14.3 Поиск дерева

Мы можем вывести дерево каталогов, ссылка на который идет в коммите. Это долж- но быть описание файлов (верхнего уровня) в нашем проекте (для конкрет- ного коммита). Используйте SHA1 хэш из строки «дерева», из списка выше. Выполните:(рис. 3.50)

```
git cat-file -p <treehash>
```



```
ghislain@ghislain-VirtualBox:~/hello$ git cat-file -t ba28612
commit
ghislain@ghislain-VirtualBox:~/hello$ git cat-file -p ba28612
tree ccc73963726c5b183edf37f1412dc9ab45bfefa
parent 77d7ccd608a8a2ab35b60cb3d8ae2f5c385c2db7
author Ghislain <ghislaindadou5297@gmail.com> 1740171208 +0300
committer Ghislain <ghislaindadou5297@gmail.com> 1740171208 +0300

Added index.html.
ghislain@ghislain-VirtualBox:~/hello$ git cat-file -p treeba28612
fatal: Not a valid object name treeba28612
ghislain@ghislain-VirtualBox:~/hello$ git cat-file -p ccc7396
040000 tree 8fe0fdef2df4e00abe06f710ddd490d657e53a21 lib
ghislain@ghislain-VirtualBox:~/hello$
```

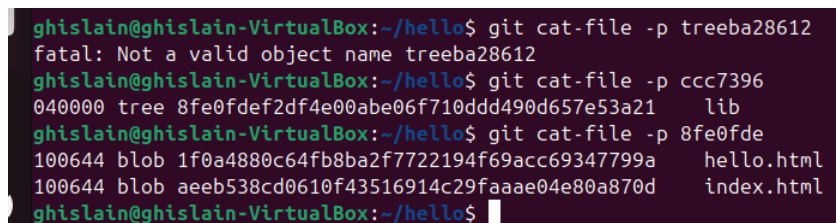
Рис. 3.50: fig64

3.14.4 Вывод каталога lib, Вывод файла hello.html

Выполните:(рис. 3.51)

```
git cat-file -p <libhash>
```

```
git cat-file -p <hellohash>
```



```
ghislain@ghislain-VirtualBox:~/hello$ git cat-file -p treeba28612
fatal: Not a valid object name treeba28612
ghislain@ghislain-VirtualBox:~/hello$ git cat-file -p ccc7396
040000 tree 8fe0fdef2df4e00abe06f710ddd490d657e53a21 lib
ghislain@ghislain-VirtualBox:~/hello$ git cat-file -p 8fe0fde
100644 blob 1f0a4880c64fb8ba2f7722194f69acc69347799a hello.html
100644 blob aeeb538cd0610f43516914c29faaae04e80a870d index.html
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.51: fig66

3.15 Создание ветки, Добавьте файл стилей style.css, Измените основную страницу и Измените index.html

(рис. 3.52), (рис. 3.53), (рис. 3.54), (рис. 3.55), (рис. 3.56), (рис. 3.57), (рис. 3.58)

```
git checkout -b style
```

```
git status
```

```
touch lib/style.css
```

Файл lib/style.css:

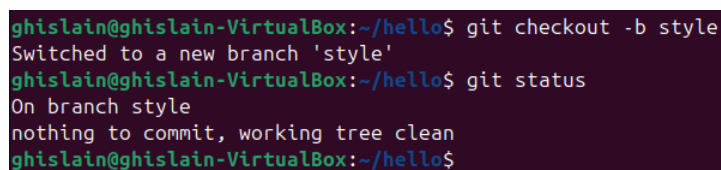
```
h1 {  
    color: red;  
}
```

```
git add lib/style.css
```

```
git commit -m "Added css stylesheet"
```

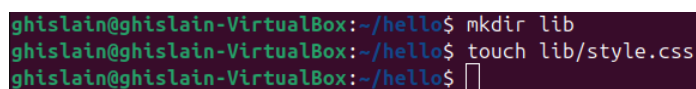
```
git add index.html
```

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




```
ghislain@ghislain-VirtualBox:~/hello$ git checkout -b style  
Switched to a new branch 'style'  
ghislain@ghislain-VirtualBox:~/hello$ git status  
On branch style  
nothing to commit, working tree clean  
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.52: fig68



```
ghislain@ghislain-VirtualBox:~/hello$ mkdir lib  
ghislain@ghislain-VirtualBox:~/hello$ touch lib/style.css  
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.53: fig69

A screenshot of a code editor window titled 'style.css' with the path '~/hello/lib'. The editor shows three lines of CSS code: '1 h1 {', '2 color: red;', and '3 }'.

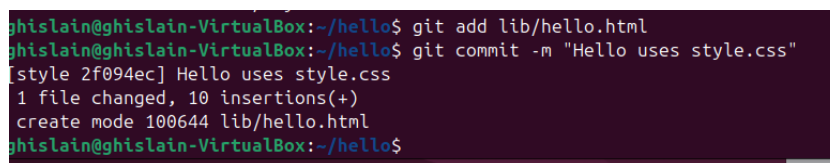
```
1 h1 {
2 color: red;
3 }
```

Рис. 3.54: fig70

A screenshot of a code editor window titled 'hello.html' with the path '~/hello/lib'. The editor shows ten lines of HTML code, including a comment, a head section with a link to style.css, and a body section with an h1 tag. A tab for 'style.css' is visible in the background.

```
1 <!-- Author: Tyem Gislen (tyemg@gmail.com) -->
2 <html>
3   <head>
4     <link type="text/css" rel="stylesheet"
5         media="all" href="style.css" />
6   </head>
7   <body>
8     <h1>Hello, World!</h1>
9   </body>
10  </html>
```

Рис. 3.55: fig72

A screenshot of a terminal window showing a series of git commands and their output. The user is in a directory ~/hello and has just added and committed a new file, lib/hello.html.

```
ghislain@ghislain-VirtualBox:~/hello$ git add lib/hello.html
ghislain@ghislain-VirtualBox:~/hello$ git commit -m "Hello uses style.css"
[style 2f094ec] Hello uses style.css
1 file changed, 10 insertions(+)
create mode 100644 lib/hello.html
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.56: fig73

A screenshot of a code editor showing ten lines of HTML code. This code uses an iframe to embed the 'hello.html' file from the 'lib' directory, with a width and height of 200 pixels.

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

Рис. 3.57: fig74

```
ghislain@ghislain-VirtualBox:~/hello/lib$ git add index.html
ghislain@ghislain-VirtualBox:~/hello/lib$ git commit -m "Updated index.html"
[style 66121f6] Updated index.html
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 lib/index.html
ghislain@ghislain-VirtualBox:~/hello/lib$
```

Рис. 3.58: fig75

3.16 Навигация по веткам

Теперь в вашем проекте есть две ветки: Выполните:(рис. 3.59)

```
git log --all
```

```
commit 66121f6dd2e28a37bdaacf34de09d4b35bdec9e9 (HEAD -> style)
Author: Ghislain <ghislaindadou5297@gmail.com>
Date: Sat Feb 22 01:44:55 2025 +0300

    Updated index.html

commit 2f094eca00ff3717d27f1fd76c756d98c64bc5de
Author: Ghislain <ghislaindadou5297@gmail.com>
Date: Sat Feb 22 01:40:58 2025 +0300

    Hello uses style.css

commit 0a508c88f2c326bc85bf2e2cc5497b72f800477f
Author: Ghislain <ghislaindadou5297@gmail.com>
Date: Sat Feb 22 01:35:40 2025 +0300

    Added css stylesheet

commit ba28612f64acc22899a5f444a60595355bfb84e8 (master)
Author: Ghislain <ghislaindadou5297@gmail.com>
Date: Fri Feb 21 23:53:28 2025 +0300

    Added index.html.

commit 77d7ccd608a8a2ab35b60cb3d8ae2f5c385c2db7
Author: Ghislain <ghislaindadou5297@gmail.com>
Date: Fri Feb 21 23:20:18 2025 +0300

    Moved hello.html to lib

commit e4333265ddbd7ecbdf00d1fed389f832094b6267
Author: Ghislain <ghislaindadou5297@gmail.com>
Date: Fri Feb 21 23:07:55 2025 +0300

:
```

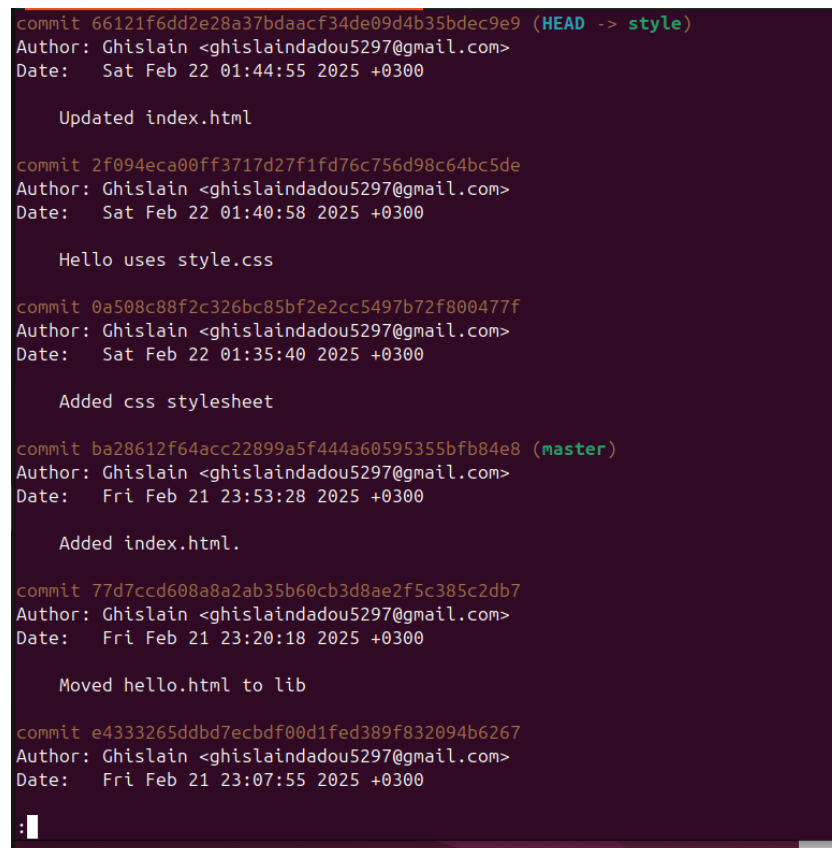
Рис. 3.59: fig76

3.16.1 Переключение на ветку master

Используйте команду `git checkout` для переключения между ветками:(рис. 3.60)

```
git checkout master
```

```
cat lib/hello.html
```



```
commit 66121f6dd2e28a37bdaacf34de09d4b35bdec9e9 (HEAD -> style)
Author: Ghislain <ghislaindadou5297@gmail.com>
Date: Sat Feb 22 01:44:55 2025 +0300

    Updated index.html

commit 2f094eca00ff3717d27f1fd76c756d98c64bc5de
Author: Ghislain <ghislaindadou5297@gmail.com>
Date: Sat Feb 22 01:40:58 2025 +0300

    Hello uses style.css

commit 0a508c88f2c326bc85bf2e2cc5497b72f800477f
Author: Ghislain <ghislaindadou5297@gmail.com>
Date: Sat Feb 22 01:35:40 2025 +0300

    Added css stylesheet

commit ba28612f64acc22899a5f444a60595355bfb84e8 (master)
Author: Ghislain <ghislaindadou5297@gmail.com>
Date: Fri Feb 21 23:53:28 2025 +0300

    Added index.html.

commit 77d7ccd608a8a2ab35b60cb3d8ae2f5c385c2db7
Author: Ghislain <ghislaindadou5297@gmail.com>
Date: Fri Feb 21 23:20:18 2025 +0300

    Moved hello.html to lib

commit e4333265ddbd7ecbdf00d1fed389f832094b6267
Author: Ghislain <ghislaindadou5297@gmail.com>
Date: Fri Feb 21 23:07:55 2025 +0300

:
```

Рис. 3.60: fig77

3.16.2 Вернемся к ветке style

Выполните:(рис. 3.61)

```
git checkout style
```

```
cat lib/hello.html
```

```
ghislain@ghislain-VirtualBox:~/hello$ git checkout style
Switched to branch 'style'
ghislain@ghislain-VirtualBox:~/hello$ cat lib/hello.html
<!-- Author: Tyem Gislen (tyemg@gmail.com) -->
<html>
<head>
  <link type="text/css" rel="stylesheet"
        media="all" href="style.css" />
</head>
<body>
  <h1>Hello, World!</h1>
</body>
</html>
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.61: fig78

3.17 Изменения в ветке master

3.17.1 Создайте файл README в ветке master(рис. 3.62)

```
git checkout master
```

```
echo "This is the Hello World example from the git tutorial." > README
```

```
ghislain@ghislain-VirtualBox:~/hello$ git checkout master
Switched to branch 'master'
ghislain@ghislain-VirtualBox:~/hello$ echo "This is the Hello World example from
the git tutorial." > README.md
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.62: fig79

3.17.2 Сделайте коммит изменений README.md в ветку master.(рис. 3.63)

```
git add README.md
```

```
git commit -m "Added README"
```

```
ghislain@ghislain-VirtualBox:~/hello$ git checkout master
Switched to branch 'master'
ghislain@ghislain-VirtualBox:~/hello$ echo "This is the Hello World example from
the git tutorial." > README.md
ghislain@ghislain-VirtualBox:~/hello$ git add README.md
ghislain@ghislain-VirtualBox:~/hello$ git commit -m "Added README"
[master 1666ee0] Added README
1 file changed, 1 insertion(+)
create mode 100644 README.md
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.63: fig80

3.18 Слияние веток

Слияние переносит изменения из двух веток в одну. Давайте вернемся к ветке style и сольем master с style. Выполните:(рис. 3.64)

```
git checkout style
git merge master
git log --graph --all
```

```
ghislain@ghislain-VirtualBox:~/hello$ git checkout style
Switched to branch 'style'
ghislain@ghislain-VirtualBox:~/hello$ git merge master
Auto-merging lib/hello.html
CONFLICT (add/add): Merge conflict in lib/hello.html
Auto-merging lib/index.html
Automatic merge failed; fix conflicts and then commit the result.
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.64: fig82

3.19 Создание конфликта

3.19.1 Вернитесь в master и создайте конфликт

Вернитесь в ветку master и внесите следующие изменения:

```
git checkout master
```

Файл lib/hello.html(рис. 3.65)

```
ghislain@ghislain-VirtualBox:~/hello$ git add lib/hello.html
ghislain@ghislain-VirtualBox:~/hello$ git commit -m 'Life is great'
[style 41be441] Life is great
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.65: fig84

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

3.20 Разрешение конфликтов

3.20.1 Слияние master с веткой style

Теперь вернемся к ветке style и попытаемся объединить ее с новой веткой master. Выполните:(рис. 3.66)

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

```
ghislain@ghislain-VirtualBox:~/hello$ git checkout style
M   lib/index.html
Already on 'style'
ghislain@ghislain-VirtualBox:~/hello$ git merge master
Already up to date.
ghislain@ghislain-VirtualBox:~/hello$
```

Рис. 3.66: fig86

Если вы откроете lib/hello.html, вы увидите:

```
<!-- Author: Dmitry S. Kulyabov (dskulyabov@rudn.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>
```

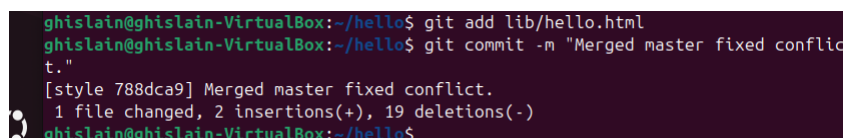
3.20.2 Решение конфликта

Вам необходимо вручную разрешить конфликт. Внесите изменения в lib/hello.html для достижения следующего результата

```
<!-- Author: Dmitry S. Kulyabov (dskulyabov@rudn.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>
```

3.20.3 Сделайте коммит решения конфликта (рис. 3.67)

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



```
ghislain@ghislain-VirtualBox:~/hello$ git add lib/hello.html
ghislain@ghislain-VirtualBox:~/hello$ git commit -m "Merged master fixed conflict."
[style 788dca9] Merged master fixed conflict.
1 file changed, 2 insertions(+), 19 deletions(-)
ghislain@ghislain-VirtualBox:~/hello$
```

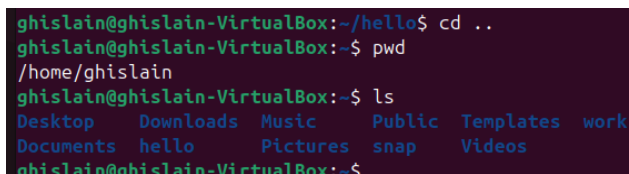
Рис. 3.67: fig86

3.21 Клонирование репозитория

3.21.1 Перейдите в рабочий каталог, Создайте клон репозитория hello и давайте взглянем на клонированный репозиторий.

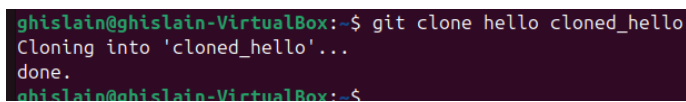
```
cd ..  
pwd  
ls  
  
git clone hello cloned_hello  
ls  
  
cd cloned_hello  
ls
```

Вы увидите список всех файлов на верхнем уровне оригинального репозитория README.md, index.html и lib.(рис. 3.68), (рис. 3.69),(рис. 3.70), (рис. 3.71)



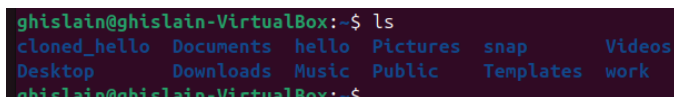
```
ghislain@ghislain-VirtualBox:~/hello$ cd ..  
ghislain@ghislain-VirtualBox:~$ pwd  
/home/ghislain  
ghislain@ghislain-VirtualBox:~$ ls  
Desktop  Downloads  Music      Public    Templates work  
Documents hello      Pictures   snap      Videos
```

Рис. 3.68: fig96



```
ghislain@ghislain-VirtualBox:~$ git clone hello cloned_hello  
Cloning into 'cloned_hello'...  
done.  
ghislain@ghislain-VirtualBox:~$
```

Рис. 3.69: fig97



```
ghislain@ghislain-VirtualBox:~$ ls  
cloned_hello Documents hello Pictures snap Videos  
Desktop Downloads Music Public Templates work  
ghislain@ghislain-VirtualBox:~$
```

Рис. 3.70: fig98

```
ghislain@ghislain-VirtualBox:~$ cd cloned_hello
ghislain@ghislain-VirtualBox:~/cloned_hello$ ls
lib  README.md
ghislain@ghislain-VirtualBox:~/cloned_hello$
```

Рис. 3.71: fig99

3.22 Что такое origin?

Выполните:

```
git remote
```

```
git remote show origin
```

Мы видим, что клонированный репозиторий знает об имени по умолчанию удаленного репозитория(рис. 3.72)

```
ghislain@ghislain-VirtualBox:~/cloned_hello$ git remote
origin
ghislain@ghislain-VirtualBox:~/cloned_hello$ git remote show origin
* remote origin
  Fetch URL: /home/ghislain/hello
  Push URL: /home/ghislain/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)
ghislain@ghislain-VirtualBox:~/cloned_hello$
```

Рис. 3.72: fig101

3.23 Извлечение общих изменений(рис. 3.73)

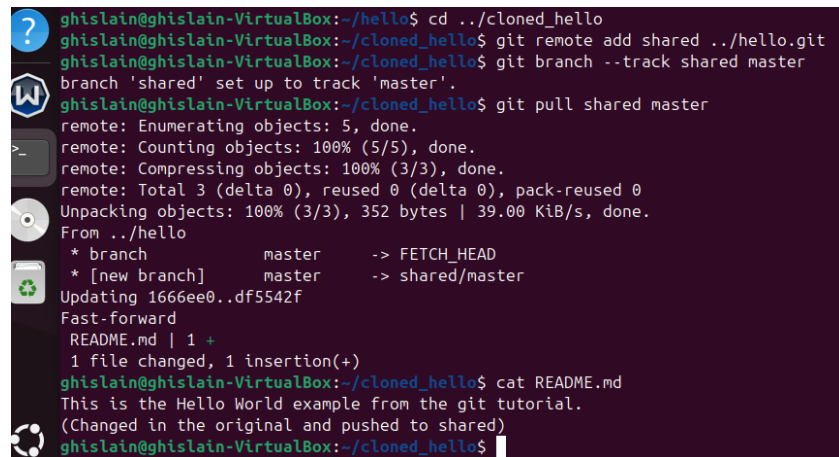
```
cd ../cloned_hello
```

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

```
git branch --track shared master
```

```
git pull shared master
```

cat README.md



```
ghislain@ghislain-VirtualBox:~/hello$ cd ../cloned_hello
ghislain@ghislain-VirtualBox:~/cloned_hello$ git remote add shared ../hello.git
ghislain@ghislain-VirtualBox:~/cloned_hello$ git branch --track shared master
branch 'shared' set up to track 'master'.
ghislain@ghislain-VirtualBox:~/cloned_hello$ git pull shared master
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
Unpacking objects: 100% (3/3), 352 bytes | 39.00 KiB/s, done.
From ../hello
* branch          master      -> FETCH_HEAD
* [new branch]    master      -> shared/master
Updating 1666ee0..df5542f
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
ghislain@ghislain-VirtualBox:~/cloned_hello$ cat README.md
This is the Hello World example from the git tutorial.
(Changed in the original and pushed to shared)
ghislain@ghislain-VirtualBox:~/cloned_hello$
```

Рис. 3.73: fig119

4 Выводы

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