

Лабораторная работа №1

Дисциплина: Математическое моделирование

Пронякова Ольга Максимовна

Содержание

1 Цель работы	5
2 Выполнение лабораторной работы	6
3 Выводы	28
Список литературы	29

Список иллюстраций

2.1 Выполнение команд	7
2.2 выполнение команд	7
2.3 Заполнение файла	8
2.4 Заполнение файла	9
2.5 Заполнение файла	10
2.6 Заполнение файла	11
2.7 Заполнение файла	11
2.8 результаты команд	12
2.9 Результат команды	13
2.10 Заполнение файла	14
2.11 результат команд	15
2.12 Результат команд	15
2.13 Результат команды	16
2.14 Изменение файла	17
2.15 Результат команды	17
2.16 Изменение файла	18
2.17 результат команды	18
2.18 Результат команды	19
2.19 Изменение файла	19
2.20 Результат команд	20
2.21 Результат команд	21
2.22 Изменение файла	21
2.23 Результат команд	22
2.24 Результат команд	22
2.25 Изменение файла	22
2.26 Результат команд	23
2.27 Результат команд	23
2.28 Результат команд	24
2.29 Результат команд	25
2.30 Изменение файла	25
2.31 Изменение файла	26
2.32 Изменение файла	26
2.33 Результат команд	27
2.34 Результат команд	27

Список таблиц

1 Цель работы

Вспомнить умения работы с git

2 Выполнение лабораторной работы

Настройка core.autocrlf с параметрами true и input делает все переводы строк текстовых файлов в главном репозитории одинаковы. 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). Если core.safecrlf установлен в true или warn, git проверяет, если преобразование является обратимым для текущей настройки core.autocrlf. core.safecrlf true - отвержение необратимого преобразования lf<->crlf. Полезно, когда специфические бинарники похожие на текстовые файлы. core.safecrlf warn - печать только предупреждение, но принимает необратимый переход.

В пустом рабочем каталоге создаю пустой каталог с именем hello, затем вхожу в него и создаю там файл с именем hello.html(рис.2.1).

```
olga@ompronyakova:~/hello$ cd hello
olga@ompronyakova:~/hello$ git init
Reinitialized existing Git repository in /home/olga/hello/.git/
olga@ompronyakova:~/hello$ git add hello.html
olga@ompronyakova:~/hello$ git commit -m "Initial commit"
[master (root-commit) da1c1a2] Initial commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 hello.html
olga@ompronyakova:~/hello$ git status
On branch master
nothing to commit, working tree clean
olga@ompronyakova:~/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")
olga@ompronyakova:~/hello$
```

Рис. 2.1: Выполнение команд

Создаю git репозиторий из этого каталога, добавляю файл в реаозиторий и проверяю статус(рис.2.2).

```
olga@ompronyakova:~/hello$ git add hello.htm
fatal: pathspec 'hello.htm' did not match any files
olga@ompronyakova:~/hello$ git add hello.html
warning: LF will be replaced by CRLF in hello.html.
The file will have its original line endings in your working directory
olga@ompronyakova:~/hello$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   hello.html
```

Рис. 2.2: выполнение команд

Изменяю содержимое файла hello.html и проверяю состояние рабочего каталога(рис.2.3).

The screenshot shows a terminal window titled "olga@olmpromyakova: ~/hello". The command "nano .git/COMMIT_EDITMSG" is run. The editor displays a template for a commit message:

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Changes to be committed:
#       modified:   hello.html
#
```

At the bottom of the screen, a menu bar for the nano editor is visible, containing options like Help, Exit, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute, Justify, Location, and Go To Line. A status bar indicates "Read 8 lines".

Рис. 2.3: Заполнение файла

Выполняю команду git, чтобы проиндексировать изменения и проверяю состояние(рис.2.4).

The screenshot shows a terminal window with the title bar "olga@ompronyakova: ~/hello". The window contains the following text:

```
GNU nano 6.2          /home/olga/hello/.git/COMMIT_EDITMSG *
Added h1 tag
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Changes to be committed:
#       modified:   hello.html
#
```

Рис. 2.4: Заполнение файла

Делаю коммит. проверяю состояние. Изменяю страницу «Hello, World», чтобы она содержала стандартные теги

и

и добавляю это изменение в индекс git(рис.2.5).

The screenshot shows a terminal window titled "olga@ompronyakova: ~/hello". The command "GNU nano 6.2" is displayed at the top left. The file content is "hello.html *", which contains the following HTML code:

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

At the bottom of the terminal, there is a menu bar with various keyboard shortcuts:

- G Help
- ^O Write Out
- ^W Where Is
- ^K Cut
- ^T Execute
- ^C Location
- X Exit
- ^R Read File
- ^A Replace
- ^U Paste
- ^J Justify
- ^/ Go To Line

Рис. 2.5: Заполнение файла

произвожу коммит проиндексированного изменения (значение по умолчанию), а затем еще раз проверяю состояние(рис.2.6).

```
olga@ompronyakova:~/hello$ nano hello.html
olga@ompronyakova:~/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

olga@ompronyakova:~/hello$ git commit -m "Added standard HTML page tags"
[master c6efe8c] Added standard HTML page tags
 1 file changed, 5 insertions(+), 1 deletion(-)
olga@ompronyakova:~/hello$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   hello.html

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

Рис. 2.6: Заполнение файла

Получаю список произведенных изменений(рис.2.7).

```
olga@ompronyakova:~/hello$ git log
commit 34e090653cf7712a7a45906191f95253ec7a9da9 (HEAD -> master)
Author: Olga1325 <pronykova05@gmail.com>
Date:   Sat Feb 22 17:55:51 2025 +0300

    Added HTML header

commit c6efe8c3f99e7a767856c3556b101bc4017ef7b0
Author: Olga1325 <pronykova05@gmail.com>
Date:   Sat Feb 22 17:54:24 2025 +0300

    Added standard HTML page tags

commit d0a78ab03436f188970cdb13876d5a9ef9872cb6
Author: Olga1325 <pronykova05@gmail.com>
Date:   Sat Feb 22 17:37:01 2025 +0300

    Added h1 tag

commit da1c1a2944bb3193cef4aefde73dfc97e61aaf18
Author: Olga1325 <pronykova05@gmail.com>
Date:   Sat Feb 22 17:23:18 2025 +0300
```

Рис. 2.7: Заполнение файла

Применяю несколько вариантов отображения лога(рис.2.8).

```
34e090653cf7712a7a45906191f95253ec7a9da9 (HEAD -> master) Added HTML header
c6efe8c3f99e7a767856c3556b101bc4017ef7b0 Added standard HTML page tags
d0a78ab03436f188970cdb13876d5a9ef9872cb6 Added h1 tag
da1c1a2944bb3193cef4aefde73dfc97e61aaf18 Initial commit
olga@ompronyakova:~/hello$ git log --pretty=oneline --max-count=2
34e090653cf7712a7a45906191f95253ec7a9da9 (HEAD -> master) Added HTML header
c6efe8c3f99e7a767856c3556b101bc4017ef7b0 Added standard HTML page tags
olga@ompronyakova:~/hello$ git log --pretty=oneline --since='5 minutes ago'
34e090653cf7712a7a45906191f95253ec7a9da9 (HEAD -> master) Added HTML header
c6efe8c3f99e7a767856c3556b101bc4017ef7b0 Added standard HTML page tags
olga@ompronyakova:~/hello$ git log --pretty=oneline --until='5 minutes ago'
d0a78ab03436f188970cdb13876d5a9ef9872cb6 Added h1 tag
da1c1a2944bb3193cef4aefde73dfc97e61aaf18 Initial commit
olga@ompronyakova:~/hello$ git log --pretty=oneline --author=Olga1325
34e090653cf7712a7a45906191f95253ec7a9da9 (HEAD -> master) Added HTML header
c6efe8c3f99e7a767856c3556b101bc4017ef7b0 Added standard HTML page tags
d0a78ab03436f188970cdb13876d5a9ef9872cb6 Added h1 tag
da1c1a2944bb3193cef4aefde73dfc97e61aaf18 Initial commit
olga@ompronyakova:~/hello$ git log --pretty=oneline --all
34e090653cf7712a7a45906191f95253ec7a9da9 (HEAD -> master) Added HTML header
c6efe8c3f99e7a767856c3556b101bc4017ef7b0 Added standard HTML page tags
d0a78ab03436f188970cdb13876d5a9ef9872cb6 Added h1 tag
da1c1a2944bb3193cef4aefde73dfc97e61aaf18 Initial commit
olga@ompronyakova:~/hello$
```

Рис. 2.8: результаты команд

Справочная страница (рис.2.9).

```
J+| olga@ompronyakova: ~/hello Q = - X
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.

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

Рис. 2.9: Результат команды

Изучаю данные лога и найдите хэш для первого коммита. Он должен быть в последней строке данных. Использую этот хэш-код (достаточно первых 7 знаков) в команде ниже. Затем проверяю содержимое файла hello.html(рис.2.10).

```
olga@ompronyakova:~/hello$ git checkout da1c1a2944bb3193cef4aefde73dfc97e61aaf18
Note: switching to 'da1c1a2944bb3193cef4aefde73dfc97e61aaf18'.

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 da1c1a2 Initial commit
olga@ompronyakova:~/hello$
```

Рис. 2.10: Заполнение файла

Теги для предыдущих версий Давайте создадим тег для версии, которая идет перед текущей версией и назовем его v1-beta. В первую очередь нам надо переключиться на предыдущую версию. Вместо поиска до хэш, мы будем использовать ^, обозначающее «родитель v1». Вместо обозначения $v1^$ можно использовать $v1~1$. Это обозначение можно определить как «первую версию предшествующую v1»(рис.2.11).

```
t:
    hello.html
Please commit your changes or stash them before you switch branches.
Aborting
olga@ompronyakova:~/hello$ cat hello.html
Hello, world
olga@ompronyakova:~/hello$ nano hello.html
olga@ompronyakova:~/hello$ nano hello.html
olga@ompronyakova:~/hello$ git checkout master
error: Your local changes to the following files would be overwritten by checkou
t:
    hello.html
Please commit your changes or stash them before you switch branches.
Aborting
olga@ompronyakova:~/hello$ cat hello.html
<html>
    <head>
    </head>
    <body>
        <h1>Hello, World!</h1>
    </body>
</html>
olga@ompronyakova:~/hello$ git tag v1
olga@ompronyakova:~/hello$
```

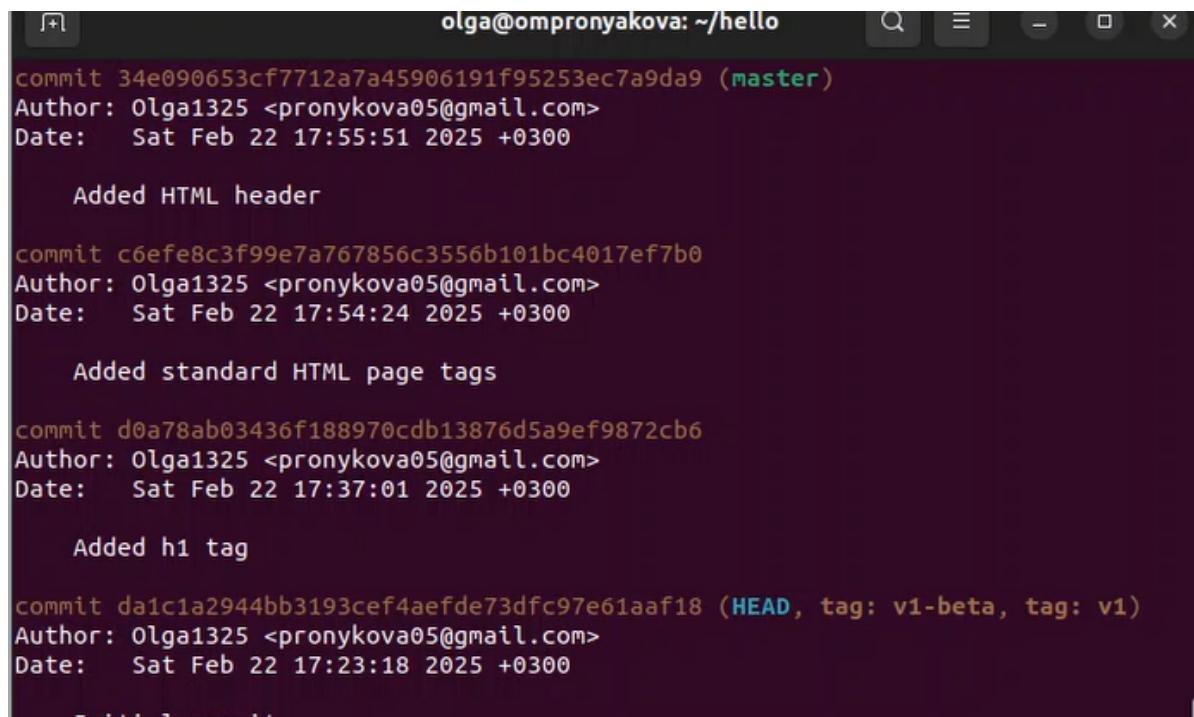
Рис. 2.11: результат команд

делаю ее версией v1-beta и пробую попереключаться между двумя отмеченными версиями(рис.2.12).

```
</html>
olga@ompronyakova:~/hello$ git tag v1
olga@ompronyakova:~/hello$ git checkout v1^
error: pathspec 'v1^' did not match any file(s) known to git
olga@ompronyakova:~/hello$ cat hello.html
<html>
    <head>
    </head>
    <body>
        <h1>Hello, World!</h1>
    </body>
</html>
olga@ompronyakova:~/hello$ git tag v1-beta
olga@ompronyakova:~/hello$ git checkout v1
M      hello.html
HEAD is now at da1c1a2 Initial commit
olga@ompronyakova:~/hello$ git checkout v1-beta
M      hello.html
HEAD is now at da1c1a2 Initial commit
olga@ompronyakova:~/hello$ git tag
v1
v1-beta
olga@ompronyakova:~/hello$
```

Рис. 2.12: Результат команд

Смотрю теги(рис.2.13).



```
olga@ompronyakova: ~/hello
commit 34e090653cf7712a7a45906191f95253ec7a9da9 (master)
Author: Olga1325 <pronykova05@gmail.com>
Date:   Sat Feb 22 17:55:51 2025 +0300

    Added HTML header

commit c6efe8c3f99e7a767856c3556b101bc4017ef7b0
Author: Olga1325 <pronykova05@gmail.com>
Date:   Sat Feb 22 17:54:24 2025 +0300

    Added standard HTML page tags

commit d0a78ab03436f188970cdb13876d5a9ef9872cb6
Author: Olga1325 <pronykova05@gmail.com>
Date:   Sat Feb 22 17:37:01 2025 +0300

    Added h1 tag

commit da1c1a2944bb3193cef4aefde73dfc97e61aaf18 (HEAD, tag: v1-beta, tag: v1)
Author: Olga1325 <pronykova05@gmail.com>
Date:   Sat Feb 22 17:23:18 2025 +0300

Initial commit
```

Рис. 2.13: Результат команды

Вношу изменение в файл hello.html в виде нежелательного комментария(рис.2.14).

```
GNU nano 6.2                                     hello.html *
```

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

```
^G Help      ^O Write Out  ^W Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File  ^\ Replace   ^U Paste    ^J Justify  ^/ Go To Line
```

Рис. 2.14: Изменение файла

Использую команду git checkout для переключения версии файла hello.html в репозитории(рис.2.15).

```
initial commit
olga@ompronyakova:~/hello$ git checkout master
error: Your local changes to the following files would be overwritten by checkout:
  hello.html
Please commit your changes or stash them before you switch branches.
Aborting
olga@ompronyakova:~/hello$ nano hello.html
olga@ompronyakova:~/hello$ git status
HEAD detached at da1c1a2
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")
olga@ompronyakova:~/hello$ git checkout hello.html
Updated 1 path from the index
```

Рис. 2.15: Результат команды

Вношу изменение в файл hello.html в виде нежелательного коммента-

рия(рис.2.16).

```
olga@ompronyakova:~/hello$ nano hello.html
GNU nano 6.2
<html>
  <head>
    <!-- This is an unwanted but staged comment -->
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>

File Name to Write: hello.html |
^C Help          M-D DOS Format      M-A Append        M-B Backup File
^C Cancel        M-M Mac Format      M-P Prepend      ^T Browse
```

Рис. 2.16: Изменение файла

Проиндексирую это изменение и проверяю состояние нежелательного изменения(рис.2.17).

```
olga@ompronyakova:~/hello$ nano hello.html
olga@ompronyakova:~/hello$ git add hello.html
warning: LF will be replaced by CRLF in hello.html.
The file will have its original line endings in your working directory
olga@ompronyakova:~/hello$ git status
HEAD detached at da1c1a2
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   hello.html
```

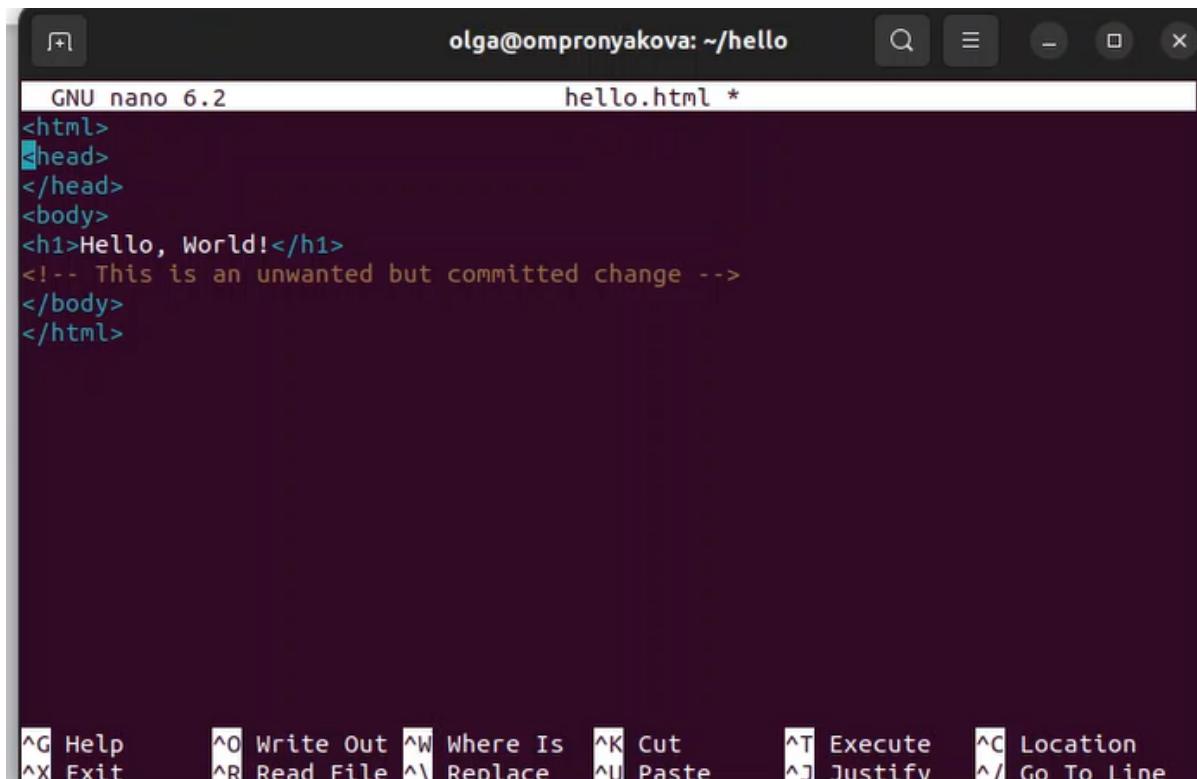
Рис. 2.17: результат команды

делаю отмену индексации изменений и проверяю статус(рис.2.18).

```
olga@ompronyakova:~/hello$ cat hello.html
<html>
    <head>
        <!-- This is an unwanted but staged comment -->
    </head>
    <body>
        <h1>Hello, World!</h1>
    </body>
</html>
olga@ompronyakova:~/hello$ git reset HEAD hello.htm
olga@ompronyakova:~/hello$ git reset HEAD hello.html
Unstaged changes after reset:
M      hello.html
olga@ompronyakova:~/hello$ git checkout hello.html
Updated 1 path from the index
olga@ompronyakova:~/hello$ git status
HEAD detached at da1c1a2
nothing to commit, working tree clean
```

Рис. 2.18: Результат команды

Изменяю файл hello.html на следующий(рис.2.19).



The screenshot shows a terminal window with the title bar "olga@ompronyakova: ~/hello". The window contains the command "GNU nano 6.2" followed by the file content "hello.html *". The content of the file is:

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

At the bottom of the terminal window, there is a menu bar with various keyboard shortcuts for nano editor commands.

Рис. 2.19: Изменение файла

Проверяю лог на нежелательные и отмененные коммиты в мой репозито-

рий(рис.2.20).

```
olga@ompronyakova:~/hello$ git log
commit 0cc699b85e720b0d117d8eadaa4191a3710f2c45 (HEAD)
Author: Olga1325 <pronykova05@gmail.com>
Date:   Sat Feb 22 18:33:53 2025 +0300

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

    This reverts commit 38b5e37a323ac3ff3d774c8a867a26eee09af7eb.

commit 38b5e37a323ac3ff3d774c8a867a26eee09af7eb
Author: Olga1325 <pronykova05@gmail.com>
Date:   Sat Feb 22 18:33:27 2025 +0300

    Oops, we didn't want this commit

commit da1c1a2944bb3193cef4aefde73dfc97e61aaf18 (tag: v1-beta, tag: v1)
Author: Olga1325 <pronykova05@gmail.com>
Date:   Sat Feb 22 17:23:18 2025 +0300

    Initial commit
olga@ompronyakova:~/hello$ git tag oops
olga@ompronyakova:~/hello$
```

Рис. 2.20: Результат команд

Но прежде чем удалить коммиты, давайте отметим последний коммит тегом, чтобы потом можно было его найти. сбросим ветку до этой точки. Поскольку ветка имеет тег, мы можем использовать имя тега в команде сброса. Тег oops свою функцию выполнил. Удаляю его и коммиты, на которые он ссылался, сборщиком мусора(рис.2.21).

```
olga@ompronyakova: ~/hello
Added HTML header

commit c6efe8c3f99e7a767856c3556b101bc4017ef7b0
Author: Olga1325 <pronykova05@gmail.com>
Date:   Sat Feb 22 17:54:24 2025 +0300

    Added standard HTML page tags

commit d0a78ab03436f188970cdb13876d5a9ef9872cb6
Author: Olga1325 <pronykova05@gmail.com>
Date:   Sat Feb 22 17:37:01 2025 +0300

    Added h1 tag

commit da1c1a2944bb3193cef4aefde73dfc97e61aaf18 (HEAD, tag: v1-beta, tag: v1)
Author: Olga1325 <pronykova05@gmail.com>
Date:   Sat Feb 22 17:23:18 2025 +0300

    Initial commit
olga@ompronyakova:~/hello$ nano hello.html
olga@ompronyakova:~/hello$ git add hello.html
warning: LF will be replaced by CRLF in hello.html.
The file will have its original line endings in your working directory
olga@ompronyakova:~/hello$
```

Рис. 2.21: Результат команд

Обновляю страницу hello, включив в нее email(рис.2.22).

```
J+I olga@ompronyakova: ~/hello
GNU nano 6.2                                     hello.html *
<!-- Author: Olga M. Pronyakova (11322226453@pfur.ru) -->
<html>
    <head>
    </head>
    <body>
        <h1>Hello, World!</h1>
    </body>
</html>
```

Рис. 2.22: Изменение файла

Действительно не хотим создавать отдельный коммит только ради электронной почты. Давайте изменим предыдущий коммит, включив в него адрес электронной почты(рис.2.23).

```
olga@ompronyakova:~/hello$ nano hello.html
olga@ompronyakova:~/hello$ git add hello.html
warning: LF will be replaced by CRLF in hello.html.
The file will have its original line endings in your working directory
olga@ompronyakova:~/hello$ git commit -m "Add an author comment"
[detached HEAD 4fdd933] Add an author comment
 1 file changed, 8 insertions(+)
olga@ompronyakova:~/hello$ nano hello.html
olga@ompronyakova:~/hello$ git add hello.html
warning: LF will be replaced by CRLF in hello.html.
The file will have its original line endings in your working directory
olga@ompronyakova:~/hello$ git commit -m "Add an author/email comment"
[detached HEAD 6db7b70] Add an author/email comment
 1 file changed, 1 insertion(+), 1 deletion(-)
olga@ompronyakova:~/hello$
```

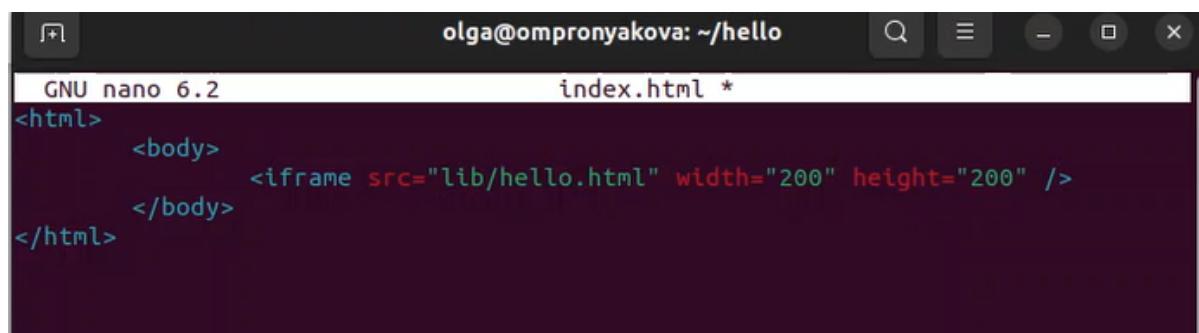
Рис. 2.23: Результат команд

Создаю структуру нашего репозитория. Давайте перенесем страницу в каталог lib(рис.2.24).

```
olga@ompronyakova:~/hello$ mkdir lib
olga@ompronyakova:~/hello$ git mv hello.html lib
olga@ompronyakova:~/hello$ git status
HEAD detached from da1c1a2
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed:   hello.html -> lib/hello.html
```

Рис. 2.24: Результат команд

Добавляю файл index.html в наш репозиторий и делаю коммит(рис.2.25).



```
olga@ompronyakova:~/hello
GNU nano 6.2          index.html *
<html>
  <body>
    <iframe src="lib/hello.html" width="200" height="200" />
  </body>
</html>
```

Рис. 2.25: Изменение файла

Каталог, в котором хранится вся информация git. здесь набор каталогов, имена которых состоят из 2 символов. Имена каталогов являются первыми двумя буквами

ми хэша sha1 объекта, хранящегося в git. Смотрим в один из каталогов с именем из 2 букв. Вы увидите файлы с именами из 38 символов. Это файлы, содержащие объекты, хранящиеся в git(рис.2.26).

```
olga@ompronyakova:~/hello$ ls -C .git
branches           config      hooks    logs      packed-refs
COMMIT_EDITMSG     description index   objects   refs
COMMIT_EDITMSG.save HEAD       info    ORIG_HEAD
olga@ompronyakova:~/hello$ ls -C .git/objects/<dir>
bash: syntax error near unexpected token `newline'
olga@ompronyakova:~/hello$ ls -C .git/objects
00  0c  23  38  4f  62  6d  a4  c6  d0  da  e8  info
02  1c  34  41  5a  65  93  b5  cf  d5  e6  f7  pack
olga@ompronyakova:~/hello$
```

Рис. 2.26: Результат команд

Узнаю файлы в подкаталоге тегов. Каждый файл соответствует тегу, ранее созданному с помощью команды git tag. Его содержание – это всего лишь хэш коммита, привязанный к тегу. Каталог heads практически аналогичен, но используется для веток, а не тегов. На данный момент у нас есть только одна ветка, так что все, что вы увидите в этом каталоге – это ветка master(рис.2.27).

```
olga@ompronyakova:~/hello$ cat .git/config
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
olga@ompronyakova:~/hello$ ls .git/refs
heads  tags
olga@ompronyakova:~/hello$ ls .git/refs/heads
master
olga@ompronyakova:~/hello$ ls .git/refs/tags
v1  v1-beta
olga@ompronyakova:~/hello$ ls .git/refs/tags/v1
.git/refs/tags/v1
olga@ompronyakova:~/hello$ cat .git/HEAD
d538c3f2514f6c7d42f5441ac2dd89bb563c6709
olga@ompronyakova:~/hello$
```

Рис. 2.27: Результат команд

Мы можем вывести дерево каталогов, ссылка на который идет в коммите. Это должно быть описание файлов (верхнего уровня) в нашем проекте (для конкретного коммита). Используйте SHA1 хэш из строки «дерева»(рис.2.28), (рис.2.29).

```
olga@ompronyakova:~/hello$ ls .git/refs/tags
v1  v1-beta
olga@ompronyakova:~/hello$ ls .git/refs/tags/v1
.git/refs/tags/v1
olga@ompronyakova:~/hello$ cat .git/HEAD
d538c3f2514f6c7d42f5441ac2dd89bb563c6709
olga@ompronyakova:~/hello$ git log --max-count=1
commit d538c3f2514f6c7d42f5441ac2dd89bb563c6709 (HEAD)
Author: Olga1325 <pronykova05@gmail.com>
Date:   Sat Feb 22 18:59:29 2025 +0300

        Added index.html.
olga@ompronyakova:~/hello$ git cat-file -t <hash>
bash: syntax error near unexpected token `newline'
olga@ompronyakova:~/hello$ git cat-file -p <hash>
bash: syntax error near unexpected token `newline'
olga@ompronyakova:~/hello$ git cat-file -p <treehash>
bash: syntax error near unexpected token `newline'
olga@ompronyakova:~/hello$ git cat-file -p <libhash>
bash: syntax error near unexpected token `newline'
olga@ompronyakova:~/hello$ git cat-file -p <hellohash>
bash: syntax error near unexpected token `newline'
olga@ompronyakova:~/hello$ ^C
```

Рис. 2.28: Результат команд

```
olga@ompronyakova: ~/hello
committer Olga1325 <pronykova05@gmail.com> 1740239969 +0300
Added index.html.
olga@ompronyakova:~/hello$ git cat-file -p 410d632f4e03de76
100644 blob cf5bc0ea78289c0be141b578ca407f1159bd25bc    index.html
040000 tree cf92d272d58b72f5462861a96d96f45c52b7d8e8    lib
olga@ompronyakova:~/hello$ git cat-file -p cf92d272d58
100644 blob c6a9ecbf2f52ffbb4ccb47bb05dbc9ee04f36fc    hello.html
olga@ompronyakova:~/hello$ git cat-file -p c6a9ecbf2f52ffb
<!-- Author: Olga M. Pronyakova (11322226453@pfur.ru) -->
<html>
    <head>
    </head>
    <body>
        <h1>Hello, World!</h1>
    </body>
</html>
olga@ompronyakova:~/hello$ git checkout -b style
Switched to a new branch 'style'
olga@ompronyakova:~/hello$ git status
On branch style
nothing to commit, working tree clean
olga@ompronyakova:~/hello$ ouch lib/style.css
```

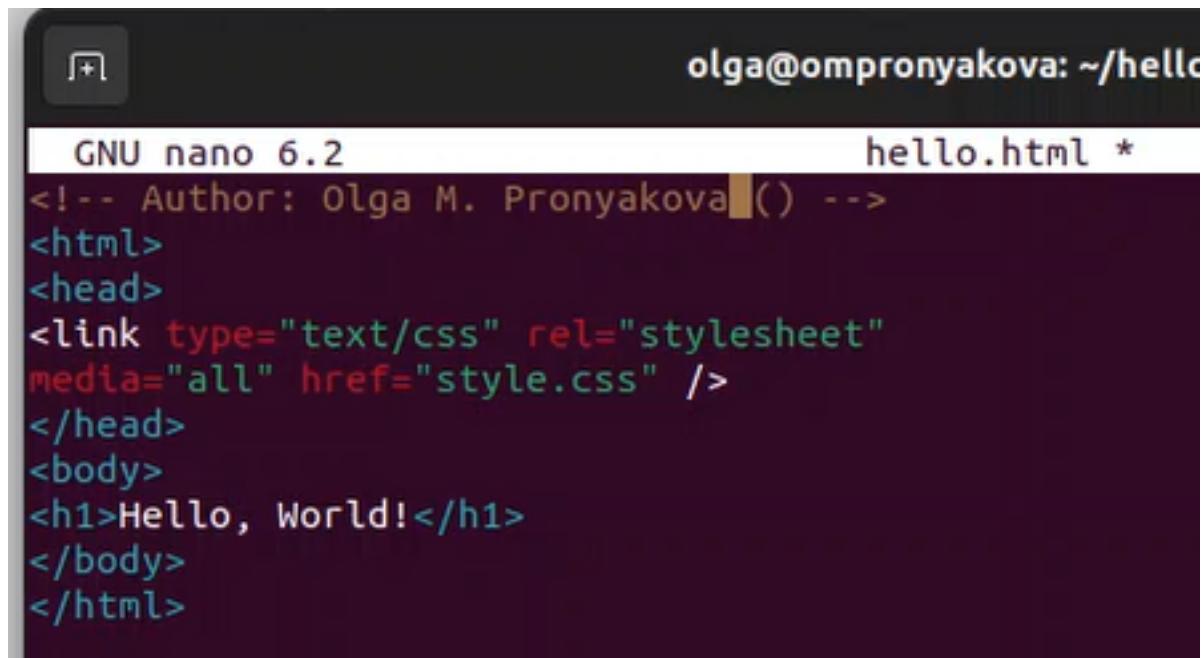
Рис. 2.29: Результат команд

создаем новую ветку «style». Добавляю файл стилей style.css(рис.2.30).

```
Open ~ /hello/lib
1 h1 {
2     color: red;
3 }
```

Рис. 2.30: Изменение файла

Обновляю файл hello.html, чтобы использовать стили style.css(рис.2.31).

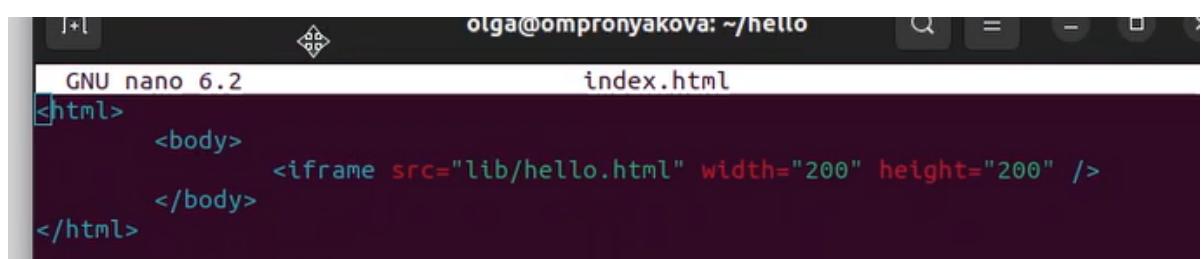


The screenshot shows a terminal window with the title bar "olga@ompronyakova: ~/hello". The main area displays the content of a file named "hello.html". The code is as follows:

```
GNU nano 6.2          hello.html *
<!-- Author: Olga M. Pronyakova -->
<html>
<head>
<link type="text/css" rel="stylesheet"
media="all" href="style.css" />
</head>
<body>
<h1>Hello, World!</h1>
</body>
</html>
```

Рис. 2.31: Изменение файла

Обновляю файл index.html, чтобы он тоже использовал style.css(рис.2.32).



The screenshot shows a terminal window with the title bar "olga@ompronyakova: ~/hello". The main area displays the content of a file named "index.html". The code is as follows:

```
GNU nano 6.2          index.html
<html>
<body>
<iframe src="lib/hello.html" width="200" height="200" />
</body>
</html>
```

Рис. 2.32: Изменение файла

Теперь у нас 2 ветки(рис.2.33), (рис.2.34).

```
olga@ompronyakova:~/hello$ git add lib/style.css
warning: LF will be replaced by CRLF in lib/style.css.
The file will have its original line endings in your working directory
olga@ompronyakova:~/hello$ git commit -m "Added css stylesheet"
[style 03cf1de] Added css stylesheet
 1 file changed, 3 insertions(+)
  create mode 100644 lib/style.css
olga@ompronyakova:~/hello$ nano hello.html
olga@ompronyakova:~/hello$ git add lib/hello.html
olga@ompronyakova:~/hello$ git commit -m "Hello uses style.css"
On branch style
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    hello.html

nothing added to commit but untracked files present (use "git add" to track)
olga@ompronyakova:~/hello$ nano index.html
olga@ompronyakova:~/hello$ git add index.html
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory
olga@ompronyakova:~/hello$ git commit -m "Updated index.html"
[style bbaf53b] Updated index.html
 1 file changed, 4 insertions(+)
olga@ompronyakova:~/hello$
```

Рис. 2.33: Результат команд

```
olga@ompronyakova:~/hello$ git log
commit bbaf53b99a073dddb05462568a67dfe8c04e0042 (HEAD -> style)
Author: Olga1325 <pronykova05@gmail.com>
Date:   Sat Feb 22 19:21:01 2025 +0300

    Updated index.html

commit 03cf1de9456e28e0ac78760b373639499f9e4fa9
Author: Olga1325 <pronykova05@gmail.com>
Date:   Sat Feb 22 19:14:58 2025 +0300

    Added css stylesheet

commit d538c3f2514f6c7d42f5441ac2dd89bb563c6709
Author: Olga1325 <pronykova05@gmail.com>
Date:   Sat Feb 22 18:59:29 2025 +0300

    Added index.html.

commit e864443cac69ac8160203e62fc2f9113606c743b
Author: Olga1325 <pronykova05@gmail.com>
Date:   Sat Feb 22 18:56:10 2025 +0300

    Moved hello.html to lib
.
```

Рис. 2.34: Результат команд

3 Выводы

Вспомнила умения работы с git

Список литературы