

Цель лабораторной работы

Изучить основные команды Git Bash, предназначенные для работы с git.

План лабораторной работы

- 1.1 Подготовка
- 1.2 Создание проекта
- 1.3 Внесение изменений
- 1.4 Индексация изменений
- 1.5 Отмена локальных изменений (до индексации)
- 1.6 Отмена проиндексированных изменений (перед коммитом)
- 1.7 Отмена коммитов
- 1.8 Удаление коммитов из ветки
- 1.9 Удаление тега oops
- 1.10 Внесение изменений в коммиты
- 1.11 Перемещение файлов
- 1.12 Второй способ перемещения файлов
- 1.13 Подробнее о структуре
- 1.14 Git внутри: Каталог .git
- 1.15 Работа непосредственно с объектами git
- 1.16 Создание ветки
- 1.17 Навигация по веткам
- 1.18 Изменения в ветке master
- 1.19 Сделать коммит изменений README.md в ветку master
- 1.20 Слияние
- 1.21 Создание конфликта
- 1.22 Разрешение конфликтов
- 1.23 Сброс ветки style
- 1.24 Сброс ветки master
- 1.25 Перебазирование
- 1.26 Слияние в ветку master
- 1.27 Клонирование репозитория
- 1.28 Просмотр клонированного репозитория
- 1.29 Что такое origin?
- 1.30 Удаленные ветки
- 1.31 Изменение оригинального репозитория
- 1.32 Слияние извлеченных изменений
- 1.33 Добавление ветки наблюдения
- 1.34 Чистые репозитории
- 1.35 Создание чистого репозитория
- 1.36 Добавление удаленного репозитория
- 1.37 Отправка изменений
- 1.38 Извлечение общих изменений

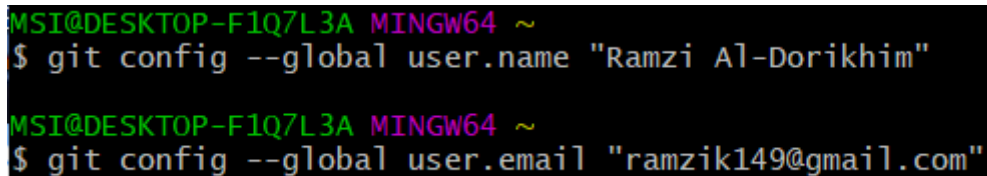
Выполнение лабораторной работы

1.1 Подготовка

1.1.1 Установка имени и электронной почты

Так как я никогда ранее не использовал git, для начала я осуществляю установку.

Передаю git свое имя и электронную почту. (рис. 001)

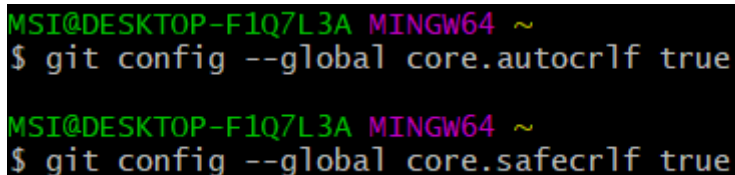


```
MSI@DESKTOP-F1Q7L3A MINGW64 ~  
$ git config --global user.name "Ramzi Al-Dorikhim"  
  
MSI@DESKTOP-F1Q7L3A MINGW64 ~  
$ git config --global user.email "ramzik149@gmail.com"
```

1.1.2 Параметры установки окончаний строк

Далее нам нужно настроить core.autocrlf, что бы все переводы строк текстовых файлов в репозитории были одинаковы.

Производим настройку для Windows. (рис. 002)

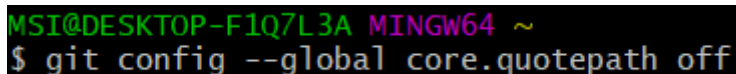


```
MSI@DESKTOP-F1Q7L3A MINGW64 ~  
$ git config --global core.autocrlf true  
  
MSI@DESKTOP-F1Q7L3A MINGW64 ~  
$ git config --global core.safecrlf true
```

1.1.3 Установка отображения unicode

По умолчанию, git будет печатать не-ASCII символов в именах файлов в виде восьмеричных последовательностей \nnn.

Что бы избежать нечитаемые строки, установим соответствующий флаг. (рис. 003)

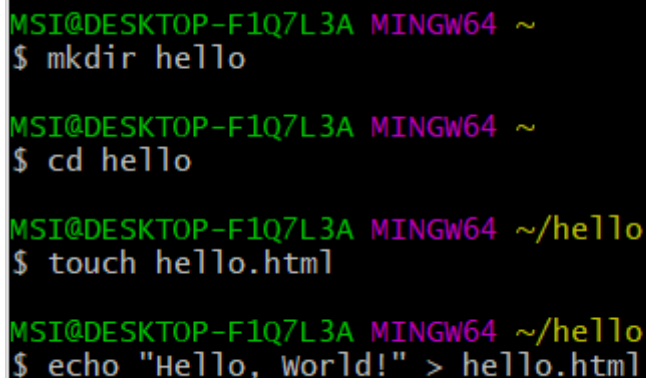


```
MSI@DESKTOP-F1Q7L3A MINGW64 ~  
$ git config --global core.quoteopath off
```

1.2 Создание проекта

1.2.1 Создаем страницу "Hello, World"

Создадим пустой каталог hello и создадим там файл с именем hello.html. (рис. 004)



```
MSI@DESKTOP-F1Q7L3A MINGW64 ~  
$ mkdir hello  
  
MSI@DESKTOP-F1Q7L3A MINGW64 ~  
$ cd hello  
  
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello  
$ touch hello.html  
  
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello  
$ echo "Hello, World!" > hello.html
```

1.2.2 Создание репозитория

Выполним команду git init, что бы создать репозиторий из этого каталога. (рис. 005)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello
$ git init
Initialized empty Git repository in C:/Users/MSI/hello/.git/
```

1.2.3 Добавление файла в репозиторий

Добавление файла в репозиторий. (рис. 006)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git add hello.html

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git commit -m "Initial Commit"
[master (root-commit) 6840ea7] Initial Commit
1 file changed, 1 insertion(+)
create mode 100644 hello.html
```

1.2.4 Проверка состояния репозитория

Используем команду git status, что бы проверить текущее состояние репозитория. (рис. 007)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git status
On branch master
nothing to commit, working tree clean
```

1.3 Внесение изменений

1.3.1 Изменение страницы "Hello, World"

Изменим содержимое файла hello.html, добавив HTML-теги. (рис. 008)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ touch hello.html

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ echo "<h1> Hello, world! </h1>" > hello.html
```

Проверим состояние рабочего каталога. (рис. 009)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ 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")
```

Git знает, что файл hello.html был изменен, но изменения еще не зафиксированы в репозитории.

1.4 Индексация изменений

Выполним команду git add <\file name> , что бы проиндексировать изменения. (рис. 010)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git add hello.html

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html
```

Изменения файла hello.html были проиндексированны. Git теперь знает об изменениях.

Правда изменение пока не записано в репозиторий.

1.4.1 Коммит изменений

Используем команду git commit, что бы сделать коммит изменений. Откроется редактор Vim, в котором я написал комментарий: "Added h1 tag". (рис. 011)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git commit
[master 5ac1631] Added h1 tag
1 file changed, 1 insertion(+), 1 deletion(-)
```

Проверим состояние каталога. (рис. 012)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git status
On branch master
nothing to commit, working tree clean
```

1.4.2 Добавим стандартные теги страницы

Изменим страницу "Hello, World", чтобы она содержала стандартные теги <html> и <body>. (рис. 013)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ touch hello.html

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ echo "<html><body><h1>Hello, world!</h1></body></html>" > hello.html
```

Добавим это изменение в индекс git и далее добавим заголовки HTML. (рис. 014)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git add hello.html

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ touch hello.html

$ echo "<html><head></head><body><h1>Hello, world!</h1></body></html>" > hello.html
```

Проверим текущий статус. (рис. 015)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ 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
```

Обратим внимание на то, что hello.html указан дважды в состоянии.

Произведём коммит проиндексированного изменения и проверим состояние еще раз. (рис. 016)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git commit -m "Added standart HTML page tags"
[master cad5684] Added standart HTML page tags
1 file changed, 1 insertion(+), 1 deletion(-)

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ 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")
```

Состояние говорит нам, что hello.html имеет незафиксированные изменения.

Теперь добавим второе изменение в индекс и проверим состояние. (рис. 017)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git add hello.html

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html
```

Второе изменение было проиндексировано и готово к коммиту.

Сделаем коммит второго изменения. (рис. 018)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git commit -m "Added HTML header"
[master acf9540] Added HTML header
1 file changed, 1 insertion(+), 1 deletion(-)
```

1.4.3 История

Получим список произведенных изменений. (рис. 019)

```
MINGW64/c/Users/MSI/hello
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git log
commit acf9540290372dd9387cb7e62efd7dfef035911d (HEAD -> master)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:56:58 2022 +0300

    Added HTML header

commit cad56849aa51a4723c6ab26abfe61cd53424142e
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:53:05 2022 +0300

    Added standart HTML page tags

commit 5ac1631fe79849dff360aac7d21ebcf05c3c1245
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:38:25 2022 +0300

    Added h1 tag

commit 6840ea78eecac12d3180226f0801d52ad7c0eb5a
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:20:21 2022 +0300

    Initial Commit

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ |
```

Однострочный формат истории. (рис. 020)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git log --pretty=oneline
acf9540290372dd9387cb7e62efd7dfef035911d (HEAD -> master) Added HTML header
cad56849aa51a4723c6ab26abfe61cd53424142e Added standart HTML page tags
5ac1631fe79849dff360aac7d21ebcf05c3c1245 Added h1 tag
6840ea78eecac12d3180226f0801d52ad7c0eb5a Initial Commit
```

1.4.4 Получение старых версий

Возвращаться назад в историю очень просто. При помощи команды checkout мы можем вернуться в любую точку.

Изучим хэши предыдущих версий, изображенных на рис. 020. Найдем хэш для первого коммита. Далее используем хэш-код 6840ea7 с командой checkout. (рис. 021)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git checkout 6840ea78
Note: switching to '6840ea78'.

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 6840ea7 Initial Commit

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello ((6840ea7...))
$ cat hello.html
Hello, World!
```

Так же на рисунке выше, мы проверили содержимое файла hello.html, после применения команды checkout. И мы видим первоначальное состояние нашего файла.

Далее вернемся к последней версии в ветке master. (рис. 022)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello ((6840ea7...))
$ git checkout master
Previous HEAD position was 6840ea7 Initial Commit
Switched to branch 'master'

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ cat hello.html
<html><head></head><body><h1>Hello, world!</h1></body></html>
```

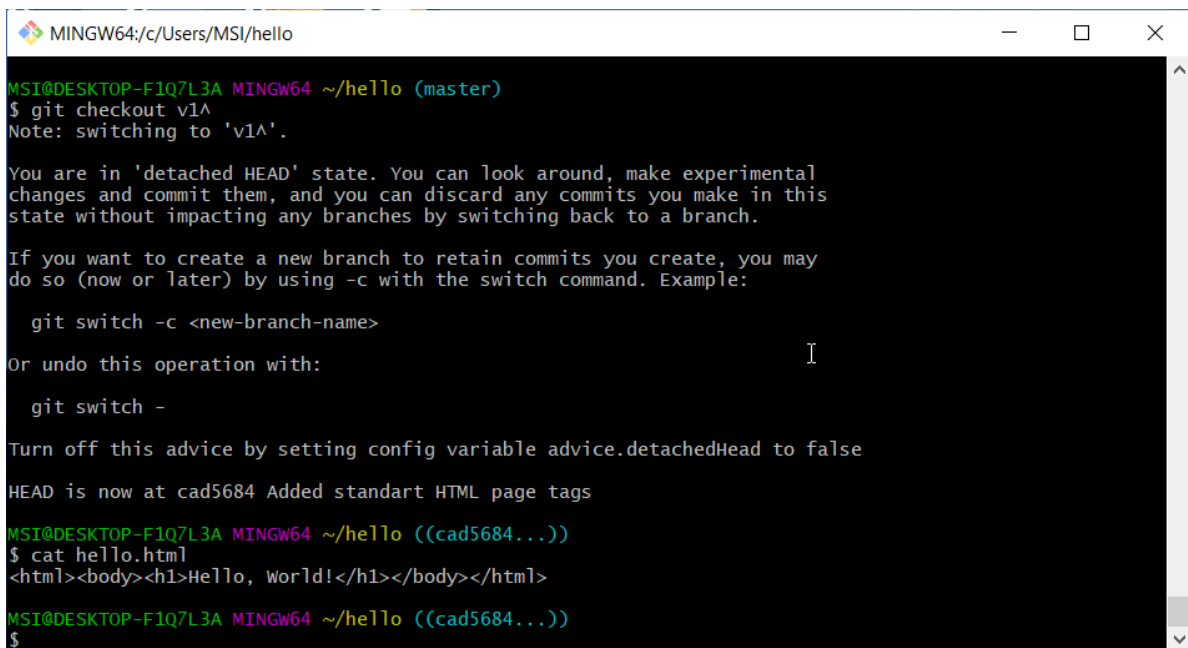
1.4.5 Создание тегов версий

Давайте назовем текущую версию страницы hello первой(v1). (рис. 023)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git tag v1
```

Теперь текущая версия страницы называется v1.

Давайте теперь создадим тег для версии, которая идет перед текущей версией и назовем его v1-beta. Для начала нам надо переключиться на предыдущую версию. Для этого используем команду `git checkout v1^`. Это должна быть версия с тегами `<html>` и `<body>`, но без `<head>`. (рис. 024)



```
MINGW64:/c/Users/MSI/hello
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ 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 cad5684 Added standart HTML page tags
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello ((cad5684...))
$ cat hello.html
<html><body><h1>Hello, world!</h1></body></html>
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello ((cad5684...))
$
```

Создание тега. (рис. 025)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello ((cad5684...))
$ git tag v1-beta
```

1.4.6 Переключение по имени тега

Теперь попробуем попереключаться между двумя отмеченными версиями. (рис. 026)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello ((v1-beta))
$ git checkout v1
Previous HEAD position was cad5684 Added standart HTML page tags
HEAD is now at acf9540 Added HTML header

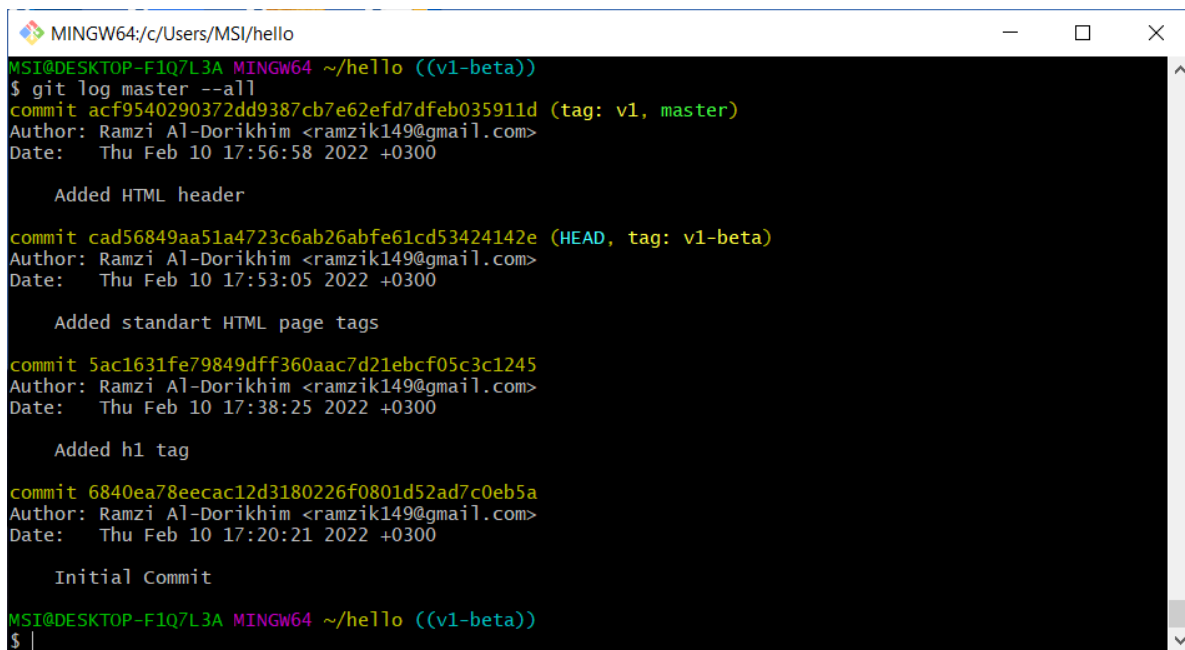
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello ((v1))
$ git checkout v1-beta
Previous HEAD position was acf9540 Added HTML header
HEAD is now at cad5684 Added standart HTML page tags
```

1.4.7 Просмотр тегов с помощью команды tag

Мы можем посмотреть наши теги. (рис. 027)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello ((v1-beta))
$ git tag
v1
v1-beta
```

Мы можем видеть теги в логе вместе с именем ветки. (рис. 028)



```
MINGW64:/c/Users/MSI/hello
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello ((v1-beta))
$ git log master --all
commit acf9540290372dd9387cb7e62efd7dfef035911d (tag: v1, master)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:56:58 2022 +0300

    Added HTML header

commit cad56849aa51a4723c6ab26abfe61cd53424142e (HEAD, tag: v1-beta)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:53:05 2022 +0300

    Added standart HTML page tags

commit 5ac1631fe79849dff360aac7d21ebcf05c3c1245
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:38:25 2022 +0300

    Added h1 tag

commit 6840ea78eacac12d3180226f0801d52ad7c0eb5a
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:20:21 2022 +0300

    Initial Commit

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello ((v1-beta))
$
```

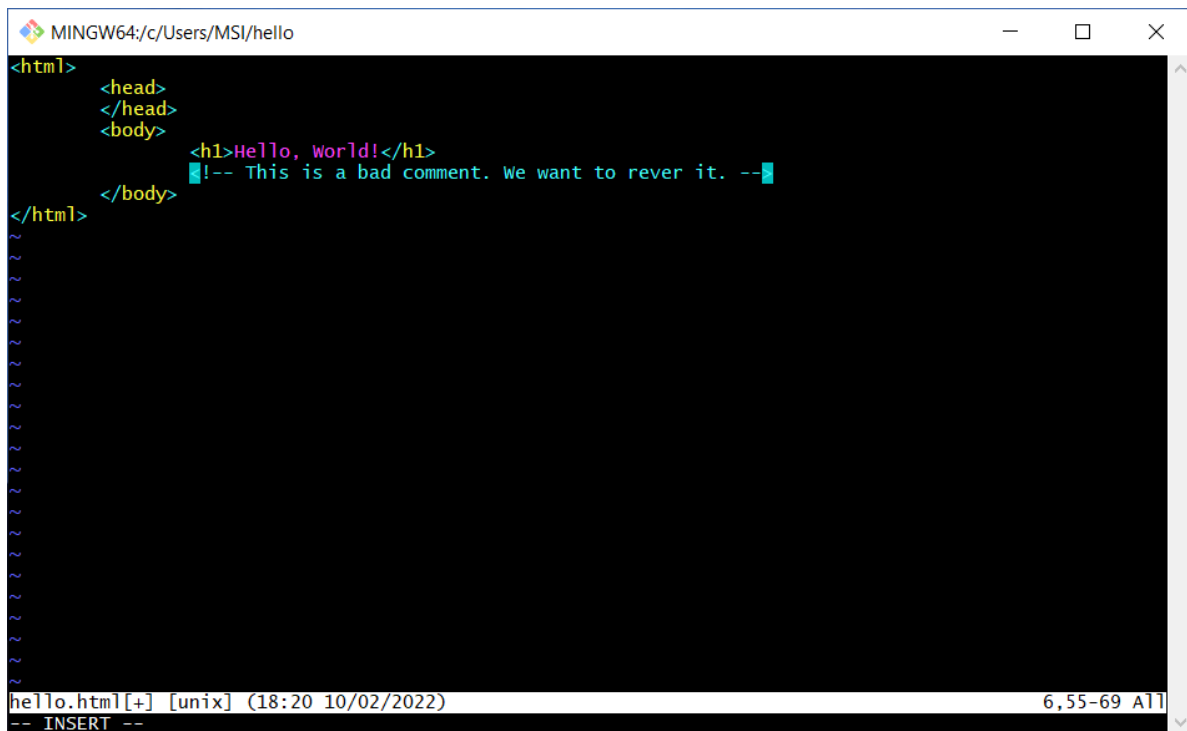
1.5 Отмена локальных изменений (до индексации)

1.5.1 Переключимся на ветку master

Убедимся, что мы находимся на последнем коммите ветки master при помощи команды `git checkout master`.

1.5.2 Изменим hello.html

Внесем нежелательные изменения в файл `hello.html`. Делать я буду это при помощи Vim. (рис. 029)



```
MINGW64:/c/Users/MSI/hello
<html>
  <head>
  </head>
  <body>
    <h1>Hello, world!</h1>
    <!-- This is a bad comment. We want to rever it. -->
  </body>
</html>
```

1.5.3 Проверим состояние

Проверим состояние рабочего каталога при помощи команды `git status`. Мы видим, что файл `hello.html` был изменен, но еще не проиндексирован. (рис. 030)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ 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")
```

1.5.4 Отменим изменения в рабочем каталоге

Используем команду `git checkout` для переключения версии файла `hello.html` в репозитории. (рис. 031)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git checkout hello.html
Updated 1 path from the index

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git status
On branch master
nothing to commit, working tree clean

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ cat hello.html
<html><head></head><body><h1>Hello, world!</h1></body></html>
```

Как мы видим, не было произведено никаких изменений, незафиксированных в рабочем каталоге.

1.6 Отмена проиндексированных изменений(перед коммитом)

1.6.1 Изменим файл и проиндексируем изменения

Внесем изменения в файл hello.html в виде нежелательного комментария. (рис. 032)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ vi hello.html

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ cat hello.html
<html>
    <head>
        <!-- This is an unwanted but staged comment -->
    </head>
    <body>
        <h1>Hello, World!</h1>
    </body>
</html>
```

1.6.2 Проверим состояние.

Проиндексируем это изменение и проверим состояние. (рис. 033)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git add hello.html

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html
```

1.6.3 Выполним сброс буферной зоны

Вывод состояния показывает нам именно то, что мы должны сделать для отмены индексации изменения: git reset HEAD hello.html. (рис. 034)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git reset HEAD hello.html
Unstaged changes after reset:
M       hello.html
```

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

1.6.4 Переключимся на версию коммита

Что бы рабочий каталог был чистым, выполним следующие команды:

git checkout hello.html

git status (рис. 035)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Наш рабочий каталог опять чист.

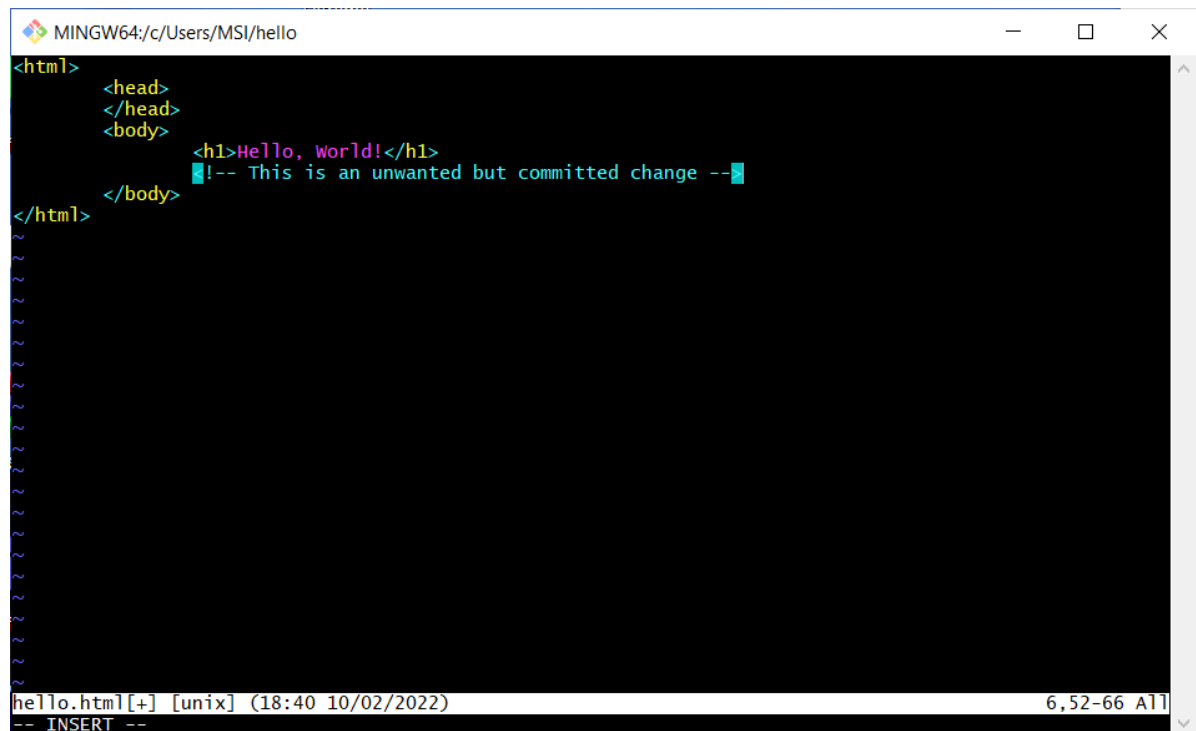
1.7 Отмена коммитов

1.7.1 Отмена коммитов

Иногда мы понимаем, что новые коммиты являются неверными. Мы отменим коммит путем создания нового коммита, отменяющего нежелательные изменения.

1.7.2 Изменим файл и сделаем коммит

Изменим файл hello.html на следующий. (рис. 036)



A screenshot of a text editor window titled 'MINGW64:/c/Users/MSI/hello'. The editor shows an HTML file with the following content:

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

The status bar at the bottom indicates 'hello.html[+] [unix] (18:40 10/02/2022)' and '6,52-66 All'.

Выполним следующие команды. (рис. 037)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ vi hello.html

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git add hello.html

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git commit -m "Oops, we didn't want this commit"
> "Oops, we didn't want this commit"
bash: t: command not found
[master 86d5d9e] Oops, we didnt want this commit
1 file changed, 8 insertions(+), 1 deletion(-)
```

1.7.3 Сделайте коммит с новыми изменениями, отменяющими предыдущие

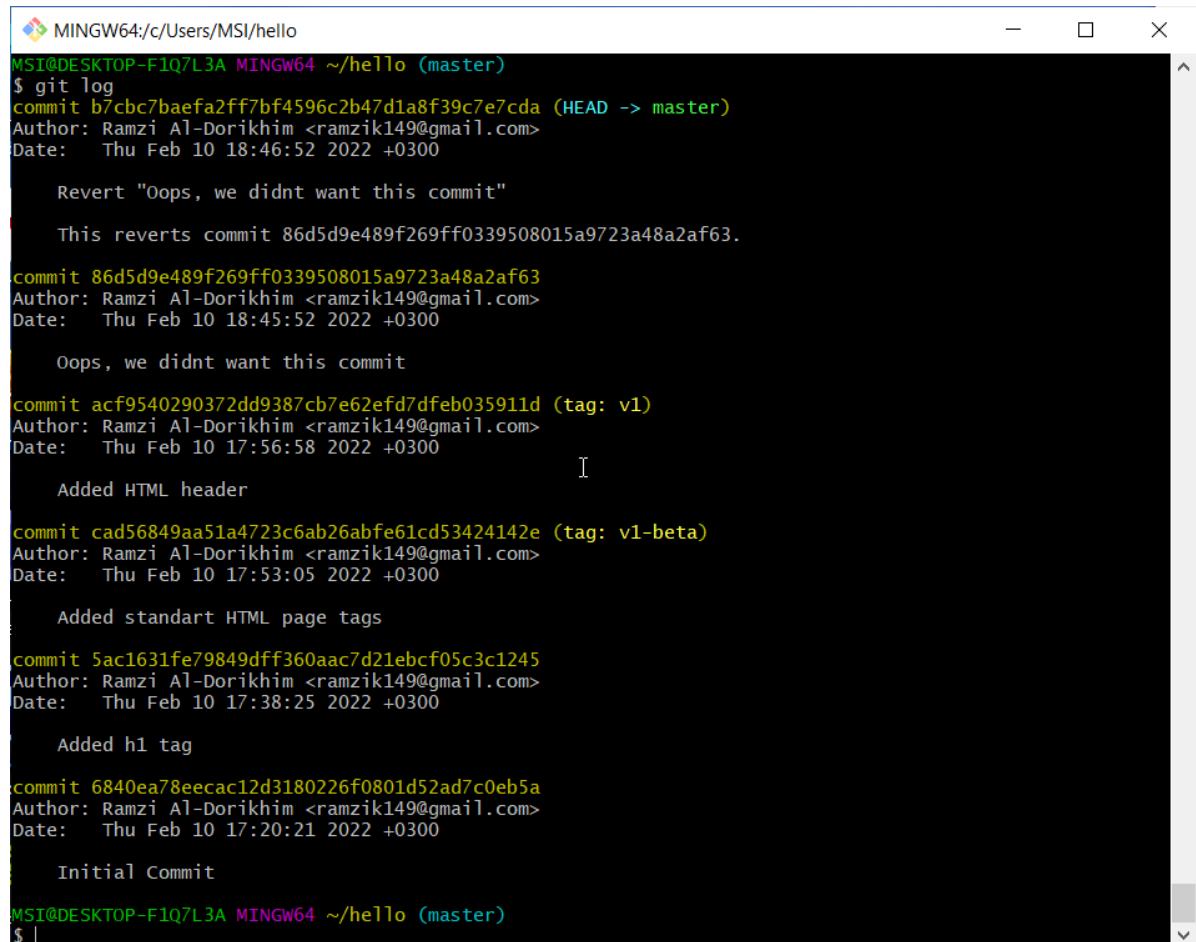
Что бы отменить коммит, нам необходимо сделать коммит, который удаляет изменения, сохраненные нежелательным коммитом. (рис. 038) (рис. 039)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git revert HEAD
[master b7cbc7b] Revert "Oops, we didnt want this commit"
1 file changed, 1 insertion(+), 8 deletions(-)
```


При получении ссылки на коммит (т.е. хэш, ветка или имя тега), команда `git reset`:
перепишет текущую ветку, чтобы она указывала на нужный коммит;
опционально сбросит буферную зону для соответствия с указанным коммитом;
опционально сбросит рабочий каталог для соответствия с указанным коммитом.

1.8.2 Проверим нашу историю

Сделаем проверку нашей истории коммитов. (рис. 041)



```
MINGW64:/c/Users/MSI/hello
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git log
commit b7cbc7baefa2ff7bf4596c2b47d1a8f39c7e7cda (HEAD -> master)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 18:46:52 2022 +0300

    Revert "Oops, we didnt want this commit"

    This reverts commit 86d5d9e489f269ff0339508015a9723a48a2af63.

commit 86d5d9e489f269ff0339508015a9723a48a2af63
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 18:45:52 2022 +0300

    Oops, we didnt want this commit

commit acf9540290372dd9387cb7e62efd7dfef035911d (tag: v1)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:56:58 2022 +0300

    Added HTML header

commit cad56849aa51a4723c6ab26abfe61cd53424142e (tag: v1-beta)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:53:05 2022 +0300

    Added standart HTML page tags

commit 5ac1631fe79849dff360aac7d21ebcf05c3c1245
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:38:25 2022 +0300

    Added h1 tag

commit 6840ea78eacac12d3180226f0801d52ad7c0eb5a
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:20:21 2022 +0300

    Initial Commit

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ |
```

Видим, что два последних коммита в этой ветке - "Oops" и "Revert Oops".

Удалим их при помощи сброса.

1.8.3 Для начала отметим эту ветку

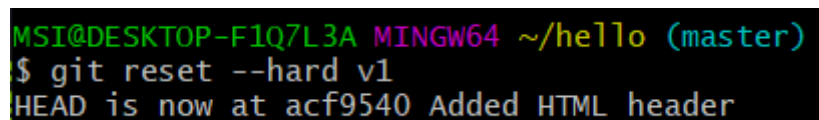
Прежде чем удалять коммиты, давайте отметим последний коммит тегом, чтобы потом можно было его найти. (рис. 042)



```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git tag oops
```

1.8.4 Сброс коммитов к предшествующим коммиту Oops

Глядя на историю лога, мы видим, что коммит с тегом «v1» является коммитом, предшествующим ошибочному коммиту. Давайте сбросим ветку до этой точки. (рис. 043)



```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git reset --hard v1
HEAD is now at acf9540 Added HTML header
```

Взглянем на log. (рис. 044)

```

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git log
commit acf9540290372dd9387cb7e62efd7dfefb035911d (HEAD -> master, tag: v1)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:56:58 2022 +0300

    Added HTML header

commit cad56849aa51a4723c6ab26abfe61cd53424142e (tag: v1-beta)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:53:05 2022 +0300

    Added standart HTML page tags

commit 5ac1631fe79849dff360aac7d21ebcf05c3c1245
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:38:25 2022 +0300

    Added h1 tag

commit 6840ea78eecac12d3180226f0801d52ad7c0eb5a
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:20:21 2022 +0300

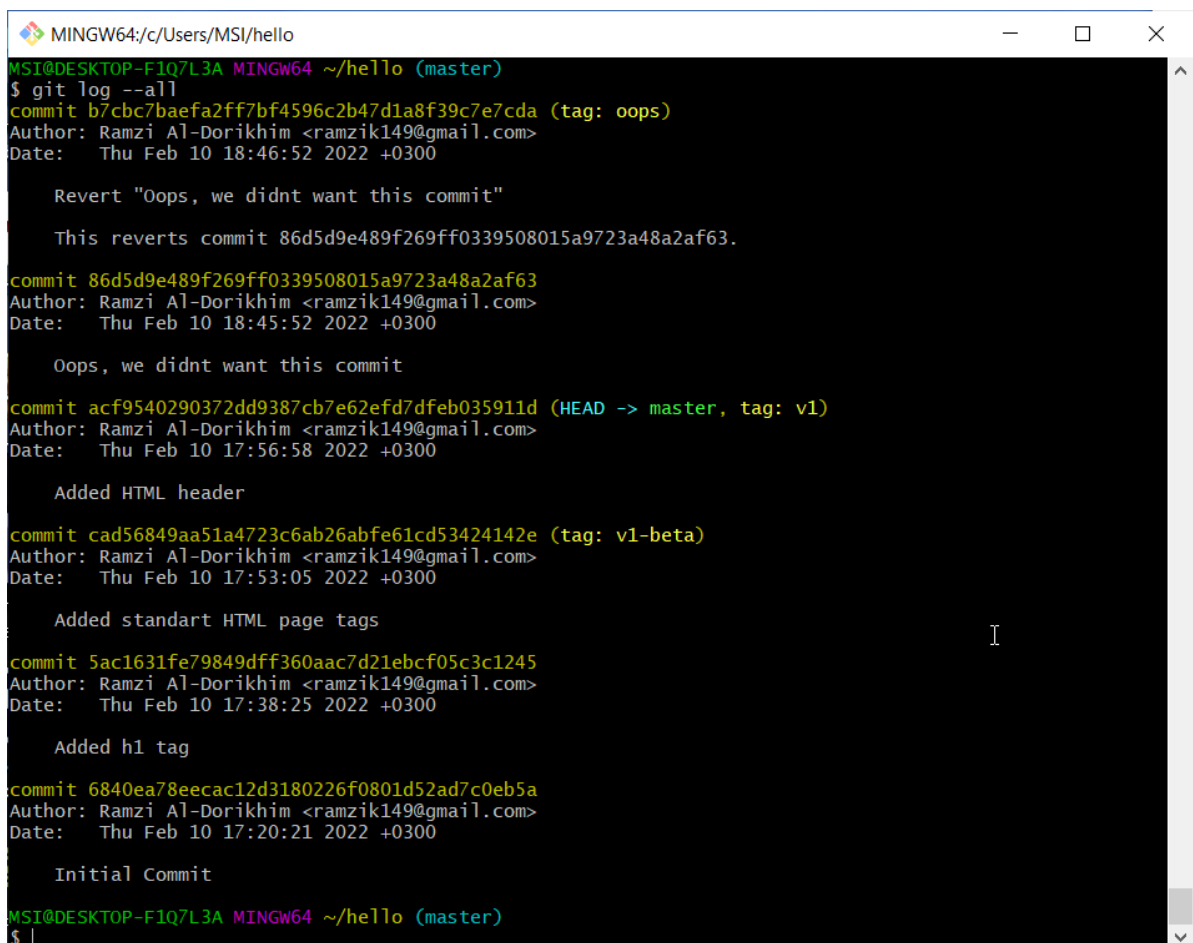
    Initial Commit

```

Наша ветка master теперь указывает на коммит v1, а коммитов Oops и Revert Oops в ветке уже нет. Параметр --hard указывает, что рабочий каталог должен быть обновлен в соответствии с новым head ветки.

1.8.5 Ничего никогда не теряется

Что же тогда случается с ошибочными коммитами? На самом деле, коммиты все еще находятся в репозитории. Мы даже можем на них ссылаться. Что бы посмотреть на все коммиты, нужно применить команду git log --all. (рис. 045)



```

MINGW64: c:/Users/MSI/hello
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git log --all
commit b7cbc7baefa2ff7bf4596c2b47d1a8f39c7e7cda (tag: oops)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 18:46:52 2022 +0300

    Revert "Oops, we didnt want this commit"

    This reverts commit 86d5d9e489f269ff0339508015a9723a48a2af63.

commit 86d5d9e489f269ff0339508015a9723a48a2af63
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 18:45:52 2022 +0300

    Oops, we didnt want this commit

commit acf9540290372dd9387cb7e62efd7dfefb035911d (HEAD -> master, tag: v1)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:56:58 2022 +0300

    Added HTML header

commit cad56849aa51a4723c6ab26abfe61cd53424142e (tag: v1-beta)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:53:05 2022 +0300

    Added standart HTML page tags

commit 5ac1631fe79849dff360aac7d21ebcf05c3c1245
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:38:25 2022 +0300

    Added h1 tag

commit 6840ea78eecac12d3180226f0801d52ad7c0eb5a
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:20:21 2022 +0300

    Initial Commit

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ |

```

Мы видим, что ошибочные коммиты не исчезли, они просто отсутствуют в ветке master.

Помните, мы отметили ошибочные коммиты тегами ? Это сделано для того, что бы была возможность на них ссылаться, не используя их хеш-коды. Коммиты, на которые нет ссылок, остаются в репозитории до тех пор, пока не будет запущен сборщик мусора.

1.8.6 Опасность сброса

Сброс в локальных ветках, как правило, безопасен. Последствия любой «аварии» как правило, можно восстановить простым сбросом с помощью нужного коммита. Однако, если ветка «расшарена» на удаленных репозиториях, сброс может сбить с толку других пользователей ветки.

1.9 Удаление тега oops

1.9.1 Удаление тега oops

Тег "oops" выполнил свою функцию. Удалим его и коммиты, на которые он ссылается при помощи сборщика мусора. (рис. 046)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git tag -d oops
Deleted tag 'oops' (was b7cbc7b)
```

Посмотрим на log. (рис. 047)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git log --all
commit acf9540290372dd9387cb7e62efd7dfefb035911d (HEAD -> master, tag: v1)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:56:58 2022 +0300

    Added HTML header

commit cad56849aa51a4723c6ab26abfe61cd53424142e (tag: v1-beta)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:53:05 2022 +0300

    Added standart HTML page tags

commit 5ac1631fe79849dff360aac7d21ebcf05c3c1245
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:38:25 2022 +0300

    Added h1 tag

commit 6840ea78eecac12d3180226f0801d52ad7c0eb5a
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:20:21 2022 +0300

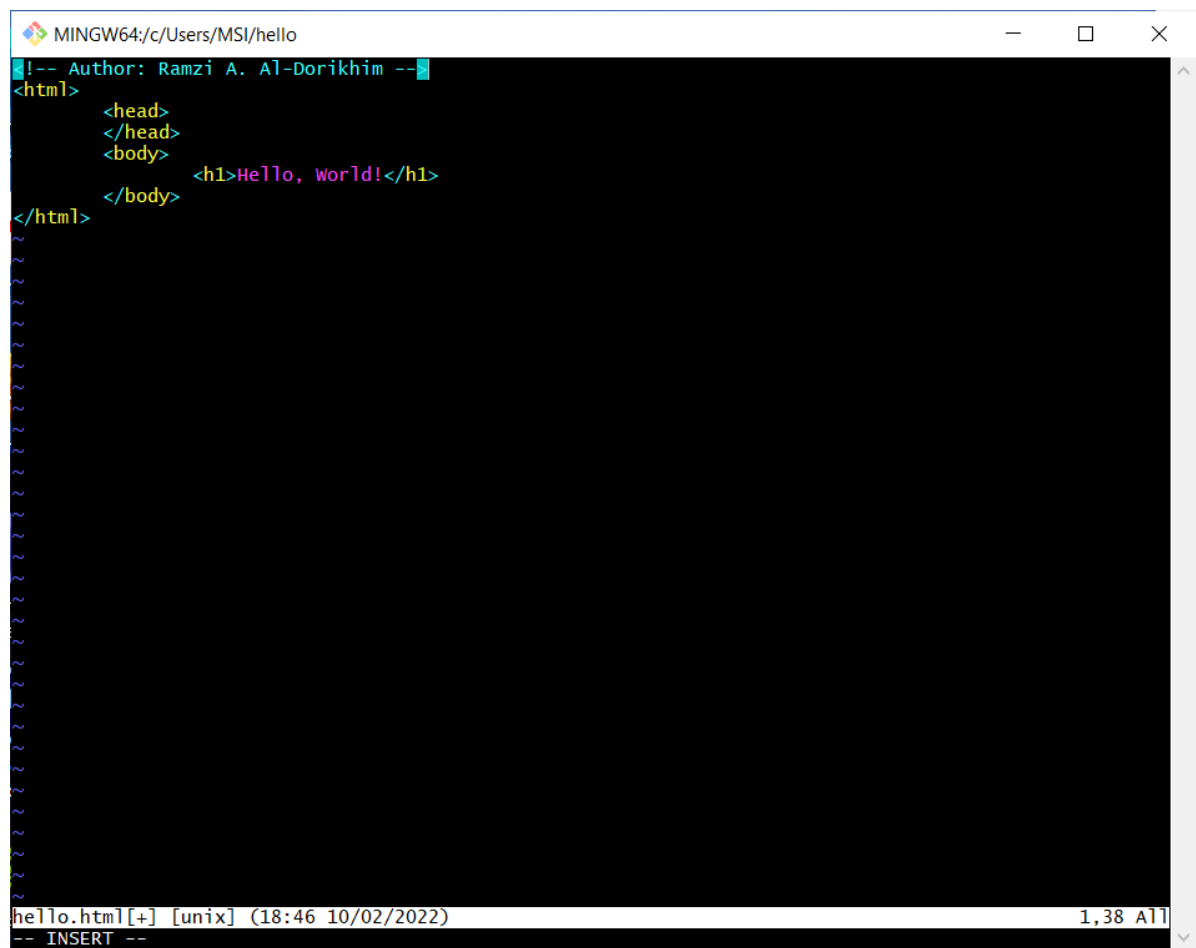
    Initial Commit
```

Тег "oops" больше не будет отображаться в репозитории.

1.10 Внесение изменений в коммиты

1.10.1 Измените страницу, а затем сделайте коммит

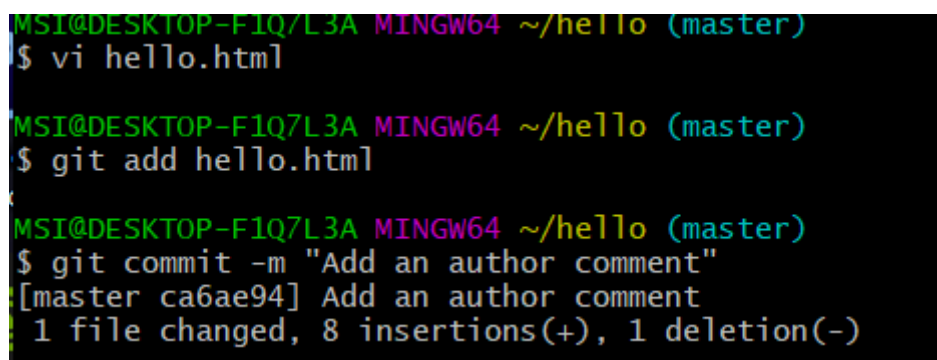
Добавим в файл hello.html свои ФИО. (рис. 048)



```
MINGW64/c/Users/MSI/hello
[!-- Author: Ramzi A. Al-Dorikhim --]
<html>
  <head>
</head>
  <body>
    <h1>Hello, world!</h1>
  </body>
</html>

hello.html[+] [unix] (18:46 10/02/2022) 1,38 All
-- INSERT --
```

Выполним. (рис. 049)



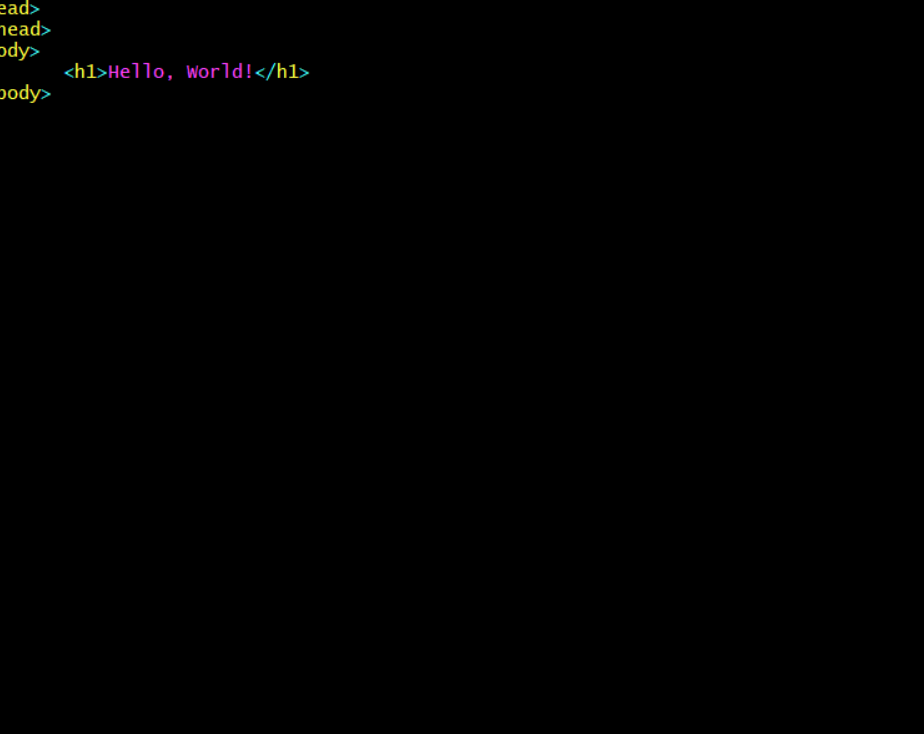
```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ vi hello.html

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git add hello.html

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git commit -m "Add an author comment"
[master ca6ae94] Add an author comment
1 file changed, 8 insertions(+), 1 deletion(-)
```

1.10.2 Необходим email

После совершения коммита мы понимаем, что любой хороший комментарий должен включать электронную почту автора. Обновим страницу hello, включив в нее email. (рис. 050)



The screenshot shows a terminal window titled "MINGW64: c:/Users/MSI/hello". The terminal content is as follows:

```
<!-- Author: Ramzi A. Al-Dorikhim (1032192882@rudn.ru) -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello, world!</h1>
  </body>
</html>
```

At the bottom of the terminal, the command prompt shows the file "hello.html" is open in a "unix" editor, with the timestamp "(18:58 10/02/2022)". The status bar at the very bottom indicates "1,56 All".

1.10.3 Изменим предыдущий коммит

Мы действительно не хотим создавать отдельный коммит только ради электронной почты. Давайте изменим предыдущий коммит, включив в него адрес электронной почты.

Выполним следующее. (рис. 051)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git add hello.html

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git commit --amend -m "Add an author/email comment"
[master 0f2a3a0] Add an author/email comment
Date: Thu Feb 10 18:59:16 2022 +0300
1 file changed, 8 insertions(+), 1 deletion(-)
```

1.10.4 Просмотр истории

Мы можем увидеть, что оригинальный коммит «автор» заменен коммитом «автор/email». Этого же эффекта можно достичь путем сброса последнего коммита в ветке, и повторного коммита новых изменений. (рис. 052)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git log
commit 0f2a3a0da23832e25fc425d918aadf809c9ac30c (HEAD -> master)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 18:59:16 2022 +0300

    Add an author/email comment

commit acf9540290372dd9387cb7e62efd7dfefb035911d (tag: v1)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:56:58 2022 +0300

    Added HTML header

commit cad56849aa51a4723c6ab26abfe61cd53424142e (tag: v1-beta)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:53:05 2022 +0300

    Added standart HTML page tags

commit 5ac1631fe79849dff360aac7d21ebcf05c3c1245
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:38:25 2022 +0300

    Added h1 tag

commit 6840ea78eecac12d3180226f0801d52ad7c0eb5a
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:20:21 2022 +0300

    Initial Commit
```

1.11 Перемещение файлов

1.11.1 Переместите файл hello.html в каталог lib

Давайте перенесем страницу в каталог lib. (рис. 053)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ mkdir lib

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git mv hello.html lib

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    hello.html -> lib/hello.html
```

Перемещая файлы с помощью git mv, мы информируем git о 2 вещах:

Что файл hello.html был удален.

Что файл lib/hello.html был создан.

Оба эти факта сразу же проиндексированы и готовы к коммиту. Команда git status сообщает, что файл был перемещен.

1.12 Второй способ перемещения файлов

Следующий набор команд идентичен нашим последним действиям. Работы здесь побольше, но результат тот же.

Мы могли бы выполнить:

```
mkdir lib
```

```
mv hello.html lib
```

```
git add lib/hello.html
```

```
git rm hello.html
```

1.12.1 Коммит в новый каталог

Сделаем коммит перемещения. (рис. 054)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git commit -m "Moved hello.html to lib"
[master 39a8957] Moved hello.html to lib
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename hello.html => lib/hello.html (100%)
```

1.13 Подробнее о структуре

1.13.1 Добавление index.html

Добавим файл index.html в наш репозиторий.(рис. 055)

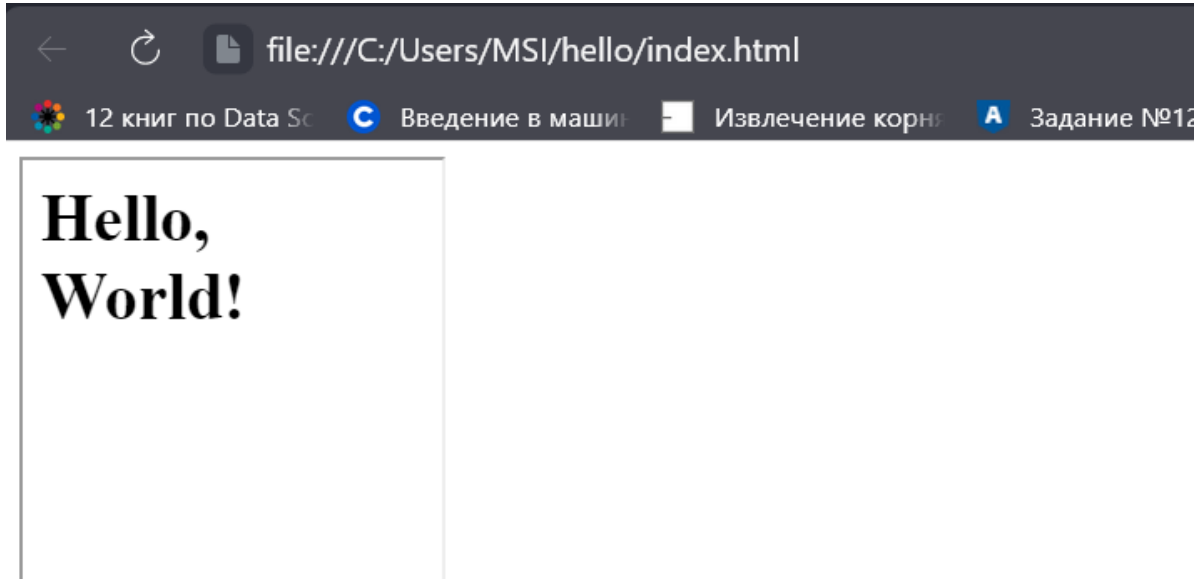


Добавим файл и сделаем коммит. (рис. 056)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git add index.html

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git commit -m "Added index.html."
[master ca32d32] Added index.html.
1 file changed, 5 insertions(+)
create mode 100644 index.html
```

Теперь при открытии index.html, мы видим кусок страницы hello в маленьком окошке. (рис. 057)



1.14 Git внутри: Каталог .git

1.14.1 Каталог .git

Выполним следующую команду. (рис. 058)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ ls -C .git
COMMIT_EDITMSG  ORIG_HEAD  description  index  logs/  packed-refs
HEAD           config     hooks/       info/  objects/  refs/
```

Это каталог, в котором хранится вся информация git.

1.14.2 База данных объектов

Выполним следующую команду. (рис. 059)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ ls -C .git/objects
0f/ 23/ 35/ 40/ 4b/ 5a/ 65/ 83/ 8a/ 97/ b7/ c8/ f5/ info/
13/ 2b/ 39/ 44/ 52/ 5d/ 68/ 86/ 8f/ ac/ bf/ ca/ f7/ pack/
```

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

1.14.3 Углубляемся в базу данных объектов

Выполним следующую команду. (рис. 060)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ ls -C .git/objects/bf
07ce307629205817a14ff174557ca925d5973f
```

Мы посмотрели в каталог bf. Мы видим файла с именами из 38 символов. Это файлы, содержащие объекты, хранящиеся в git. Они сжаты и закодированы.

1.14.4 Config File

Выполним следующую команду. (рис. 061)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true
```

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

1.14.5 Ветки и теги

Выполним следующие команды. (рис. 062)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ ls .git/refs
heads/  tags/

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ ls .git/refs/heads
master

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ ls .git/refs/tags
v1  v1-beta

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ ls .git/refs/tags/v1
.git/refs/tags/v1
```

Каждый файл соответствует тегу, ранее созданному с помощью команды git tag. Его содержание — это всего лишь хэш коммита, привязанный к тегу.

1.14.6 Файл HEAD

Выполним следующую команду. (рис. 063)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ cat .git/HEAD
ref: refs/heads/master
```

Файл HEAD содержит ссылку на текущую ветку.

1.15 Работа непосредственно с объектами git

1.15.1 Поиск последнего коммита

Выполним следующую команду. (рис. 064)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git log --max-count=1
commit ca32d321743babdce514ff01c366316ec39a3c08 (HEAD -> master)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 19:24:12 2022 +0300

    Added index.html.
```

Эта команда должна показать последний коммит в репозиторий.

1.15.2 Вывод последнего коммита с помощью SHA1 хэша

Выполним следующие команды. (рис. 065)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git cat-file -t ca32d32
commit

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git cat-file -p ca32d32
tree 35d9f085d5e5a35534cc8ea068ecb77fad4f4795
parent 39a89573b6c21978a0349951d48eafe3b57de830
author Ramzi Al-Dorikhim <ramzik149@gmail.com> 1644510252 +0300
committer Ramzi Al-Dorikhim <ramzik149@gmail.com> 1644510252 +0300

Added index.html.
```

1.15.3 Поиск дерева

Выполним следующую команду. (рис. 066)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git cat-file -p 35d9f085d5e5a35534cc8ea068ecb77fad4f4795
100644 blob 83f6e3486b747d70fd938d53c2b8df1a80b8c02b    index.html
040000 tree 5dbe4c9fc69cce8dc8bc81f979f849dd7344bbe8    lib
```

Мы можем вывести дерево каталогов, ссылка на который идет в коммите.

1.15.4 Вывод каталога lib

Выполним следующую команду. (рис. 067)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git cat-file -p 5dbe4c9fc69cce8dc8bc81f979f849dd7344bbe8
100644 blob 40e6e128e2b0baa1ecae221e2def41c9df896c2e    hello.html
```

1.15.5 Вывод файла hello.html

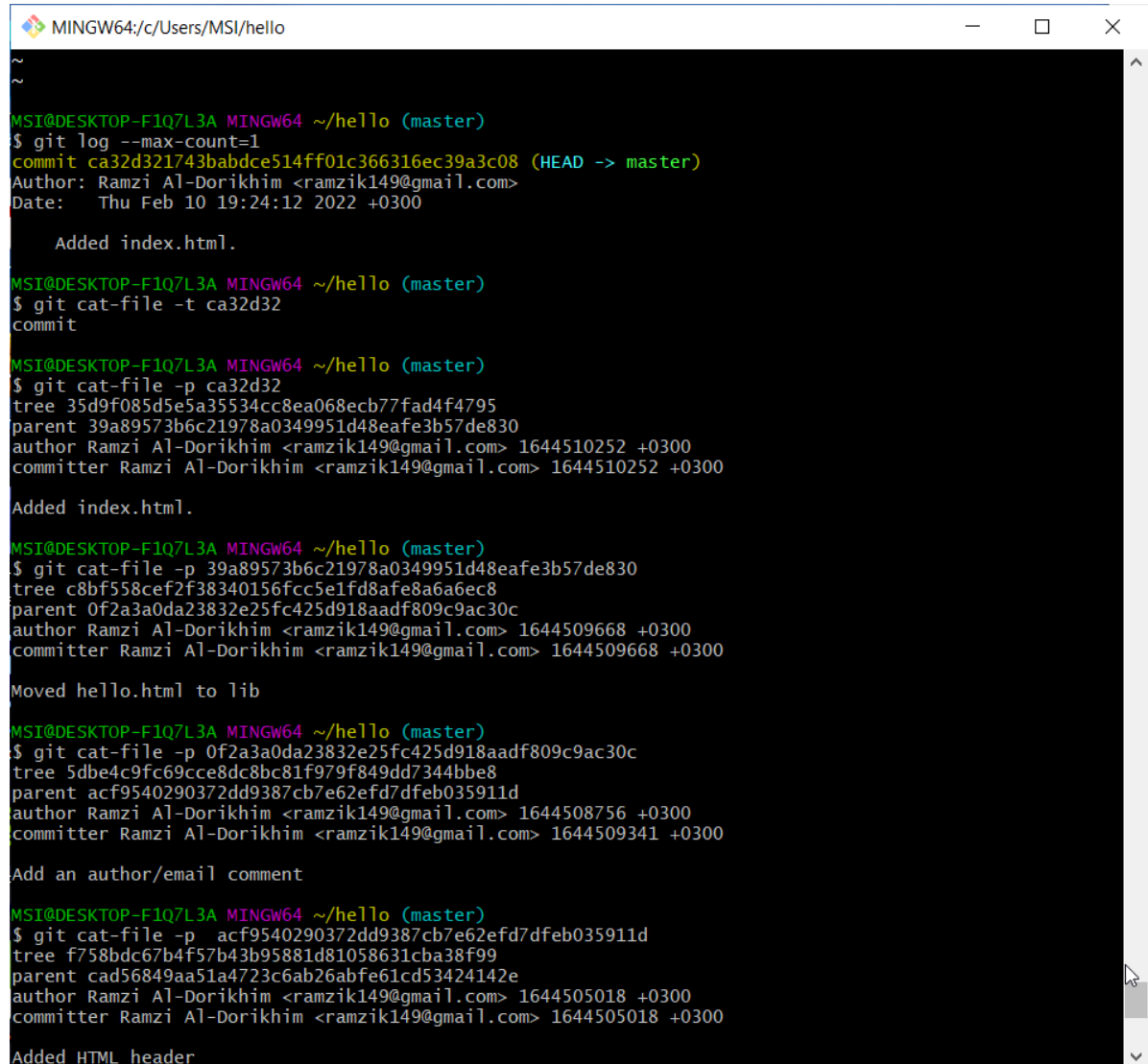
Выполним следующую команду. (рис. 068)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git cat-file -p 40e6e128e2b0baa1ecae221e2def41c9df896c2e
<!-- Author: Ramzi A. Al-Dorikhim (1032192882@rudn.ru) -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello, world!</h1>
  </body>
</html>
```

1.15.6 Исследуем самостоятельно

Исследуем git репозиторий вручную самостоятельно. Посмотрим, удастся ли нам найти оригинальный файл hello.html с самого первого коммита вручную по ссылкам SHA1 хэша в последнем коммите.

Давайте взглянем на рис. 069 и рис. 070. Я попробовал дойти с последнего коммита до самого первого по ссылкам SHA.



```
MINGW64:/c/Users/MSI/hello

~
~

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git log --max-count=1
commit ca32d321743babdce514ff01c366316ec39a3c08 (HEAD -> master)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 19:24:12 2022 +0300

    Added index.html.

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git cat-file -t ca32d32
commit

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git cat-file -p ca32d32
tree 35d9f085d5e5a35534cc8ea068ecb77fad4f4795
parent 39a89573b6c21978a0349951d48eafe3b57de830
author Ramzi Al-Dorikhim <ramzik149@gmail.com> 1644510252 +0300
committer Ramzi Al-Dorikhim <ramzik149@gmail.com> 1644510252 +0300

    Added index.html.

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git cat-file -p 39a89573b6c21978a0349951d48eafe3b57de830
tree c8bf558cef2f38340156fcc5e1fd8afe8a6a6ec8
parent 0f2a3a0da23832e25fc425d918aadf809c9ac30c
author Ramzi Al-Dorikhim <ramzik149@gmail.com> 1644509668 +0300
committer Ramzi Al-Dorikhim <ramzik149@gmail.com> 1644509668 +0300

    Moved hello.html to lib

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git cat-file -p 0f2a3a0da23832e25fc425d918aadf809c9ac30c
tree 5dbe4c9fc69cce8dc8bc81f979f849dd7344bbe8
parent acf9540290372dd9387cb7e62efd7dfef035911d
author Ramzi Al-Dorikhim <ramzik149@gmail.com> 1644508756 +0300
committer Ramzi Al-Dorikhim <ramzik149@gmail.com> 1644509341 +0300

    Add an author/email comment

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git cat-file -p acf9540290372dd9387cb7e62efd7dfef035911d
tree f758bdc67b4f57b43b95881d81058631cba38f99
parent cad56849aa51a4723c6ab26abfe61cd53424142e
author Ramzi Al-Dorikhim <ramzik149@gmail.com> 1644505018 +0300
committer Ramzi Al-Dorikhim <ramzik149@gmail.com> 1644505018 +0300

    Added HTML header
```

```

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git cat-file -p cad56849aa51a4723c6ab26abfe61cd53424142e
tree 8f14088ab9203a7d27f2a3c21c15abf20e353af3
parent 5ac1631fe79849dff360aac7d21ebcf05c3c1245
author Ramzi Al-Dorikhim <ramzik149@gmail.com> 1644504785 +0300
committer Ramzi Al-Dorikhim <ramzik149@gmail.com> 1644504785 +0300

Added standart HTML page tags

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git cat-file -p 5ac1631
tree bf07ce307629205817a14ff174557ca925d5973f
parent 6840ea78eecac12d3180226f0801d52ad7c0eb5a
author Ramzi Al-Dorikhim <ramzik149@gmail.com> 1644503905 +0300
committer Ramzi Al-Dorikhim <ramzik149@gmail.com> 1644503905 +0300

Added h1 tag

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git cat-file -p 6840ea7
tree 523cd318fc3b74ad3852cd34a74720ad6f7dd422
author Ramzi Al-Dorikhim <ramzik149@gmail.com> 1644502821 +0300
committer Ramzi Al-Dorikhim <ramzik149@gmail.com> 1644502821 +0300

Initial Commit

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git cat-file -p 523cd318fc3b74ad3852cd34a74720ad6f7dd422
100644 blob 8ab686eafeb1f44702738c8b0f24f2567c36da6d    hello.html

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git cat-file -p 8ab686eafeb1f44702738c8b0f24f2567c36da6d
Hello, World!

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$

```

Как мы видим внизу на рис. 070 , мне удалось дойти до самого первого коммита.

1.16 Создание ветки

Что бы сделать hello world более выразительным, мы создадим отдельную ветку, чтобы изолировать эти изменения от изменений в ветке master.

1.16.1 Создайте ветку

Давайте назовем нашу новую ветку «style».

Выполним следующую команду. (рис. 071)

```

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git checkout -b style
Switched to a new branch 'style'

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git status
On branch style
nothing to commit, working tree clean

```

1.16.2 Добавим файл стилей style.css

Выполним следующие команды. (рис. 072)


```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ touch lib/style.css

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ echo "h1 {color: red;}" > style.css
```

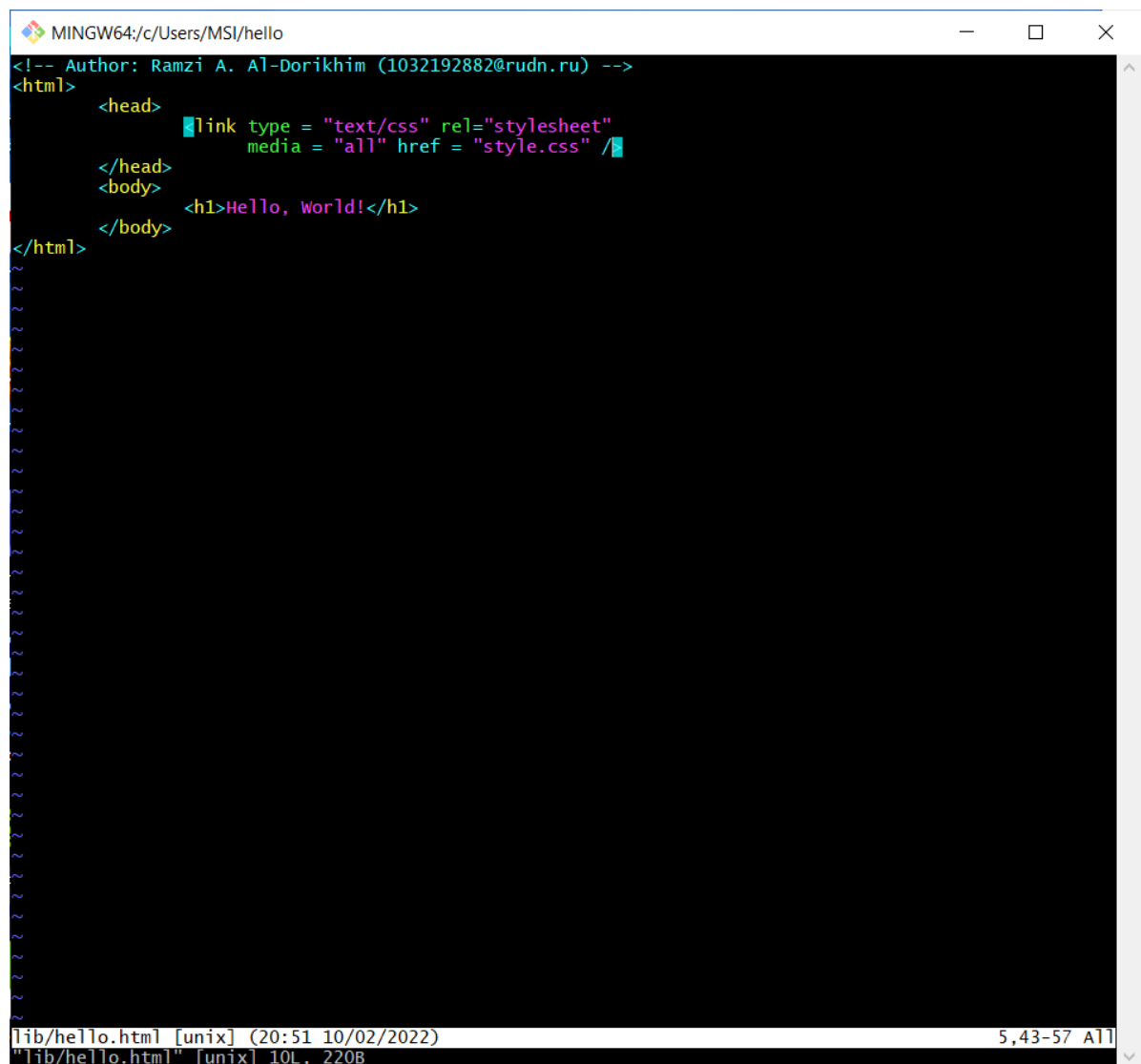
Выполним следующие команды. (рис. 073)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git add lib/style.css

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git commit -m "Added css stylesheet"
[style 5e9b631] Added css stylesheet
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 lib/style.css
```

1.16.3 Изменим основную страницу

Обновим файл hello.html, чтобы использовать стили style.css.(рис. 074)



```
MINGW64:/c/Users/MSI/hello
<!-- Author: Ramzi A. Al-Dorikhim (1032192882@rudn.ru) -->
<html>
  <head>
    <link type = "text/css" rel="stylesheet"
          media = "all" href = "style.css" />
  </head>
  <body>
    <h1>Hello, world!</h1>
  </body>
</html>
```

lib/hello.html [unix] (20:51 10/02/2022) 5,43-57 All
"lib/hello.html" [unix] 10L, 220B

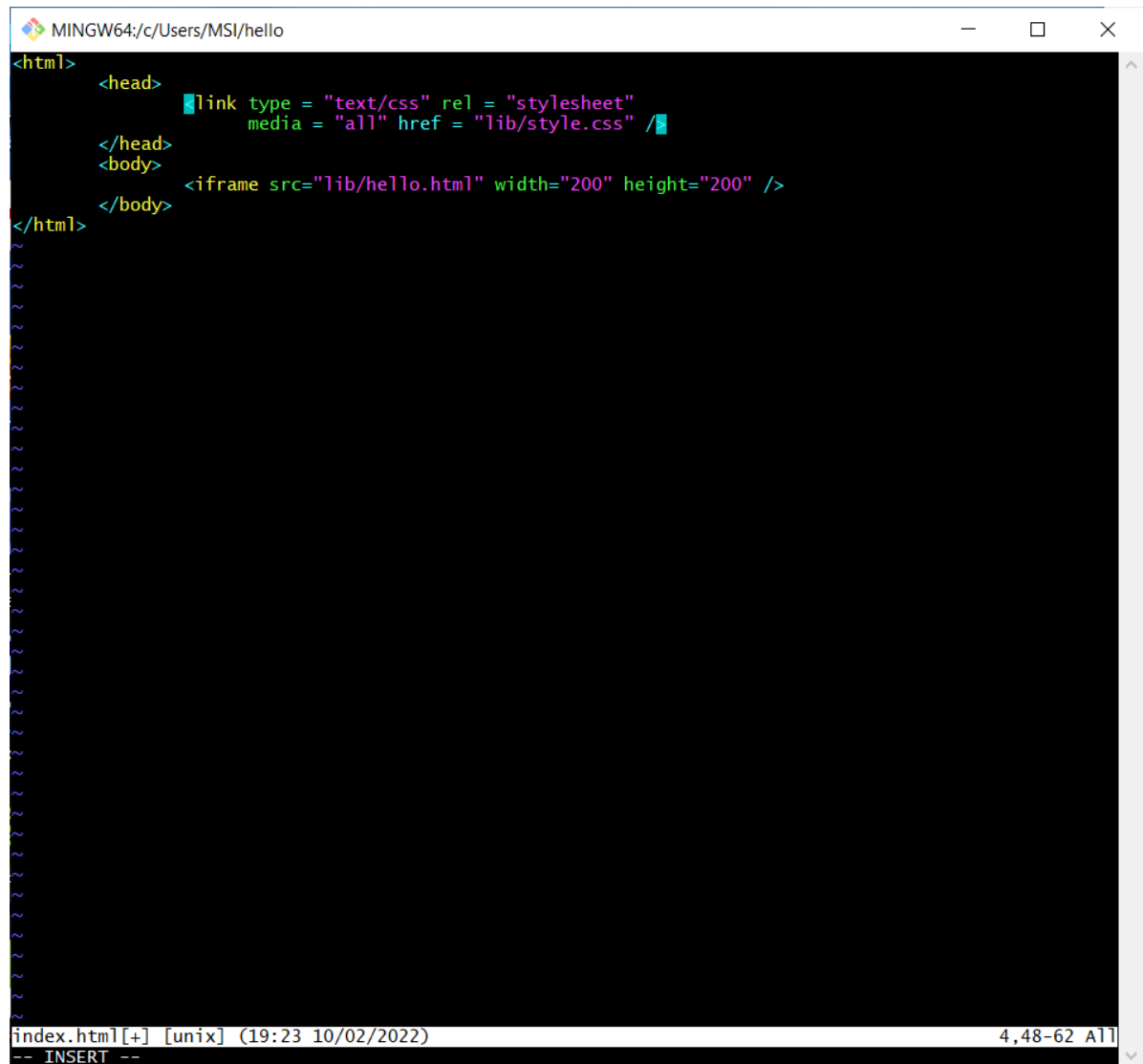
Выполним следующие команды. (рис. 075)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git add lib/hello.html

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git commit -m "Hello uses style.css"
[style 6ca361e] Hello uses style.css
1 file changed, 2 insertions(+)
```

1.16.4 Изменим index.html

Обновим файл index.html, чтобы он тоже использовал style.css (рис. 076)



```
MINGW64:/c/Users/MSI/hello
<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>
```

index.html[+] [unix] (19:23 10/02/2022) 4,48-62 All
-- INSERT --

Выполним следующие команды. (рис. 077)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git add index.html

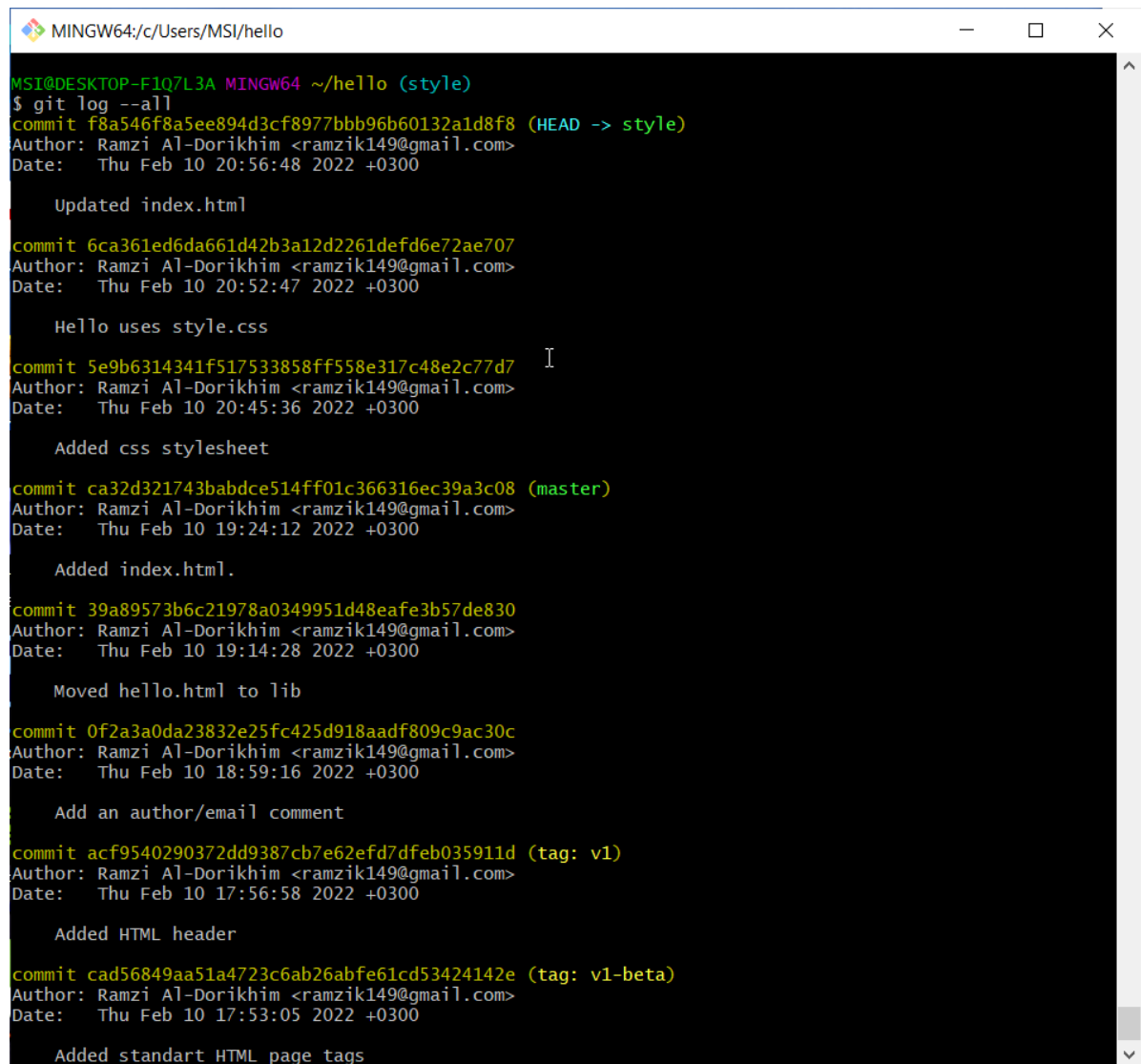
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git commit -m "Updated index.html"
[style f8a546f] Updated index.html
1 file changed, 4 insertions(+)
```

Результат. (рис. 078)

Hello, World!

1.17 Навигация по веткам

Теперь в нашем проекте есть две ветки. Выполним следующую команду. (рис. 079)



```
MINGW64: c:/Users/MSI/hello

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git log --all
commit f8a546f8a5ee894d3cf8977bbb96b60132a1d8f8 (HEAD -> style)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 20:56:48 2022 +0300

    Updated index.html

commit 6ca361ed6da661d42b3a12d2261defd6e72ae707
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 20:52:47 2022 +0300

    Hello uses style.css

commit 5e9b6314341f517533858ff558e317c48e2c77d7
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 20:45:36 2022 +0300

    Added css stylesheet

commit ca32d321743babdce514ff01c366316ec39a3c08 (master)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 19:24:12 2022 +0300

    Added index.html.

commit 39a89573b6c21978a0349951d48eafe3b57de830
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 19:14:28 2022 +0300

    Moved hello.html to lib

commit 0f2a3a0da23832e25fc425d918aadf809c9ac30c
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 18:59:16 2022 +0300

    Add an author/email comment

commit acf9540290372dd9387cb7e62efd7dfef035911d (tag: v1)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:56:58 2022 +0300

    Added HTML header

commit cad56849aa51a4723c6ab26abfe61cd53424142e (tag: v1-beta)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:53:05 2022 +0300

    Added standart HTML page tags
```

1.17.1 Переключение на ветку master

Выполним следующие команды. (рис. 080)

```

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git checkout master
Switched to branch 'master'

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ cat lib/hello.html
<!-- Author: Ramzi A. Al-Dorikhim (1032192882@rudn.ru) -->
<html>
    <head>
    </head>
    <body>
        <h1>Hello, World!</h1>
    </body>
</html>

```

Используя команду git checkout для переключения между ветками, мы переключили на ветку master.

1.17.2 Вернемся к ветке style

Выполним следующие команды. (рис. 081)

```

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git checkout style
Switched to branch 'style'

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ cat lib/hello.html
<!-- Author: Ramzi A. Al-Dorikhim (1032192882@rudn.ru) -->
<html>
    <head>
        <link type = "text/css" rel="stylesheet"
            media = "all" href = "style.css" />
    </head>
    <body>
        <h1>Hello, World!</h1>
    </body>
</html>

```

Содержимое lib/hello.html подтверждает, что мы вернулись на ветку style.

1.18 Изменения в ветке master

1.18.1 Создайте файл README в ветке master

Выполним следующие команды. (рис. 082)

```

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git checkout master
Switched to branch 'master'

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ touch README.md

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ echo "This is the Hello World example from the git tutorial." > README.md

```

Мы создали файл README.md

1.19 Сделайте коммит изменений README.md в ветку master.

Выполним следующие команды. (рис. 083)

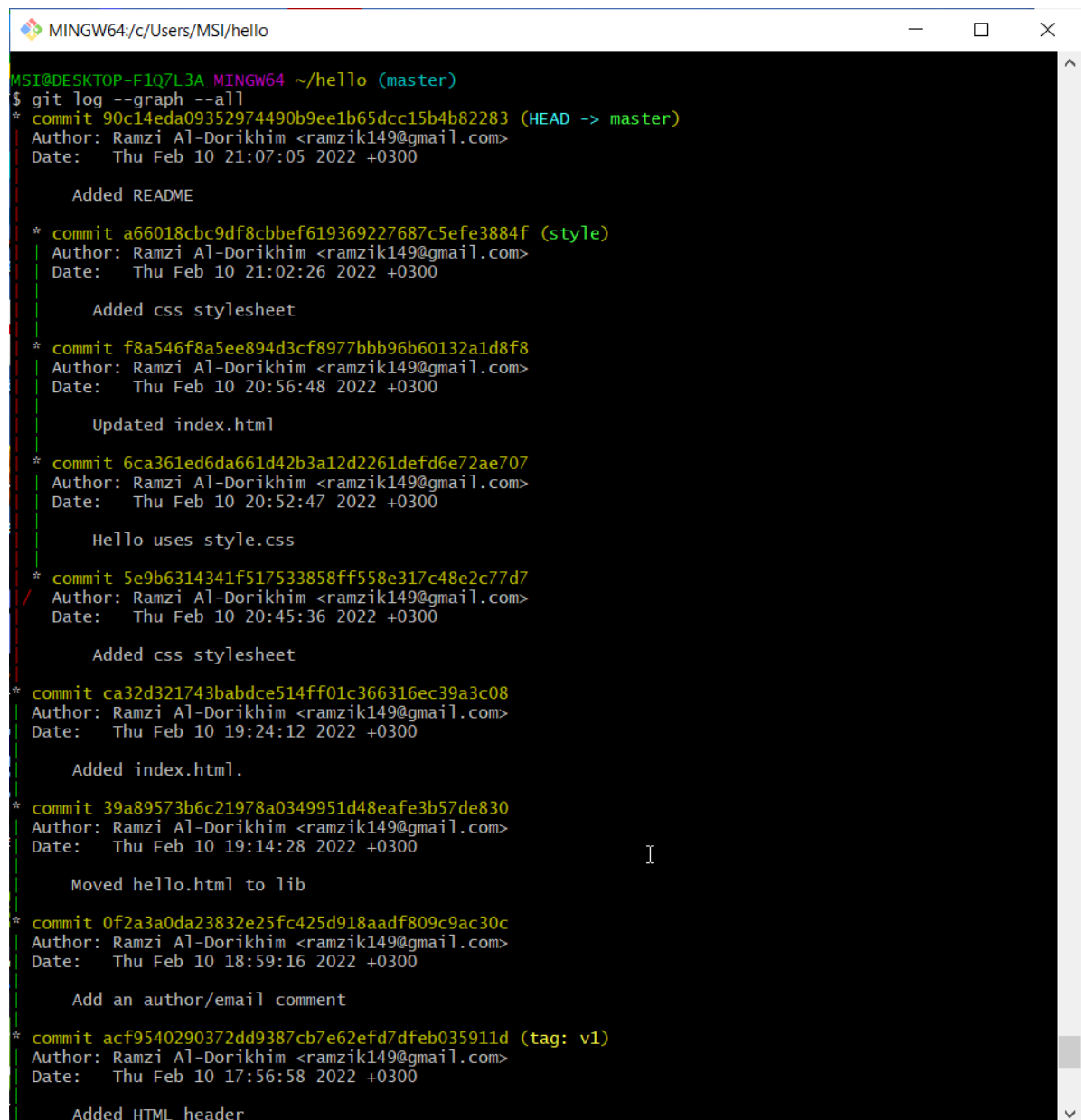
```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git add README.md

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git commit -m "Added README"
[master 90c14ed] Added README
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

1.19.1 Просмотр отличающихся веток

Теперь у нас в репозитории есть две отличающиеся ветки. Используем следующую лог-команду для просмотра веток и их отличий.

Выполним следующую команду. (рис. 084)



```
MINGW64: c:/Users/MSI/hello
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git log --graph --all
* commit 90c14eda09352974490b9ee1b65dcc15b4b82283 (HEAD -> master)
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 21:07:05 2022 +0300
    Added README

* commit a66018cbc9df8cbbef619369227687c5efe3884f (style)
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 21:02:26 2022 +0300
    Added css stylesheet

* commit f8a546f8a5ee894d3cf8977bbb96b60132a1d8f8
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 20:56:48 2022 +0300
    Updated index.html

* commit 6ca361ed6da661d42b3a12d2261defd6e72ae707
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 20:52:47 2022 +0300
    Hello uses style.css

* commit 5e9b6314341f517533858ff558e317c48e2c77d7
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 20:45:36 2022 +0300
    Added css stylesheet

* commit ca32d321743babdce514ff01c366316ec39a3c08
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 19:24:12 2022 +0300
    Added index.html.

* commit 39a89573b6c21978a0349951d48eafe3b57de830
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 19:14:28 2022 +0300
    Moved hello.html to lib

* commit 0f2a3a0da23832e25fc425d918aadf809c9ac30c
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 18:59:16 2022 +0300
    Add an author/email comment

* commit acf9540290372dd9387cb7e62efd7dfef035911d (tag: v1)
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 17:56:58 2022 +0300
    Added HTML header
```

Добавление опции --graph в git log вызывает построение дерева коммитов с помощью простых ASCII символов.

1.20 Слияние

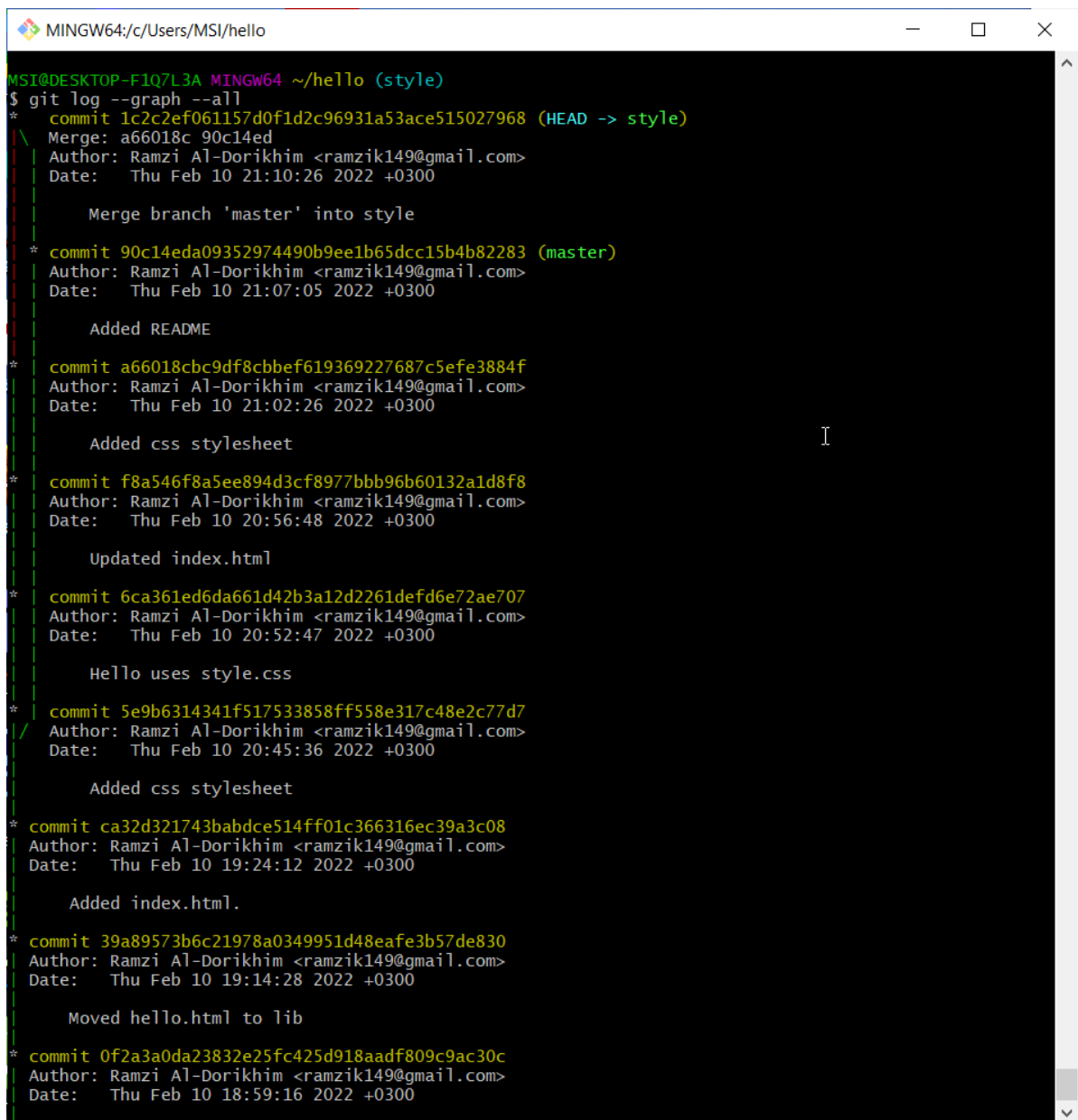
1.20.1 Слияние веток

Слияние переносит изменения из двух веток в одну. Давайте вернемся к ветке style и сольем master с style.

Выполним следующие команды. (рис. 085) (рис. 086)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git checkout style
Switched to branch 'style'

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git merge master
Merge made by the 'ort' strategy.
 README.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

A screenshot of a Windows terminal window titled "MINGW64:/c/Users/MSI/hello". The terminal shows the output of the command "git log --graph --all". The log displays a series of commits in a vertical sequence, connected by dashed lines. The commits include: a merge of master into style, adding README, adding css stylesheet, updating index.html, adding css stylesheet, adding index.html, and moving hello.html to lib. The terminal text is as follows:

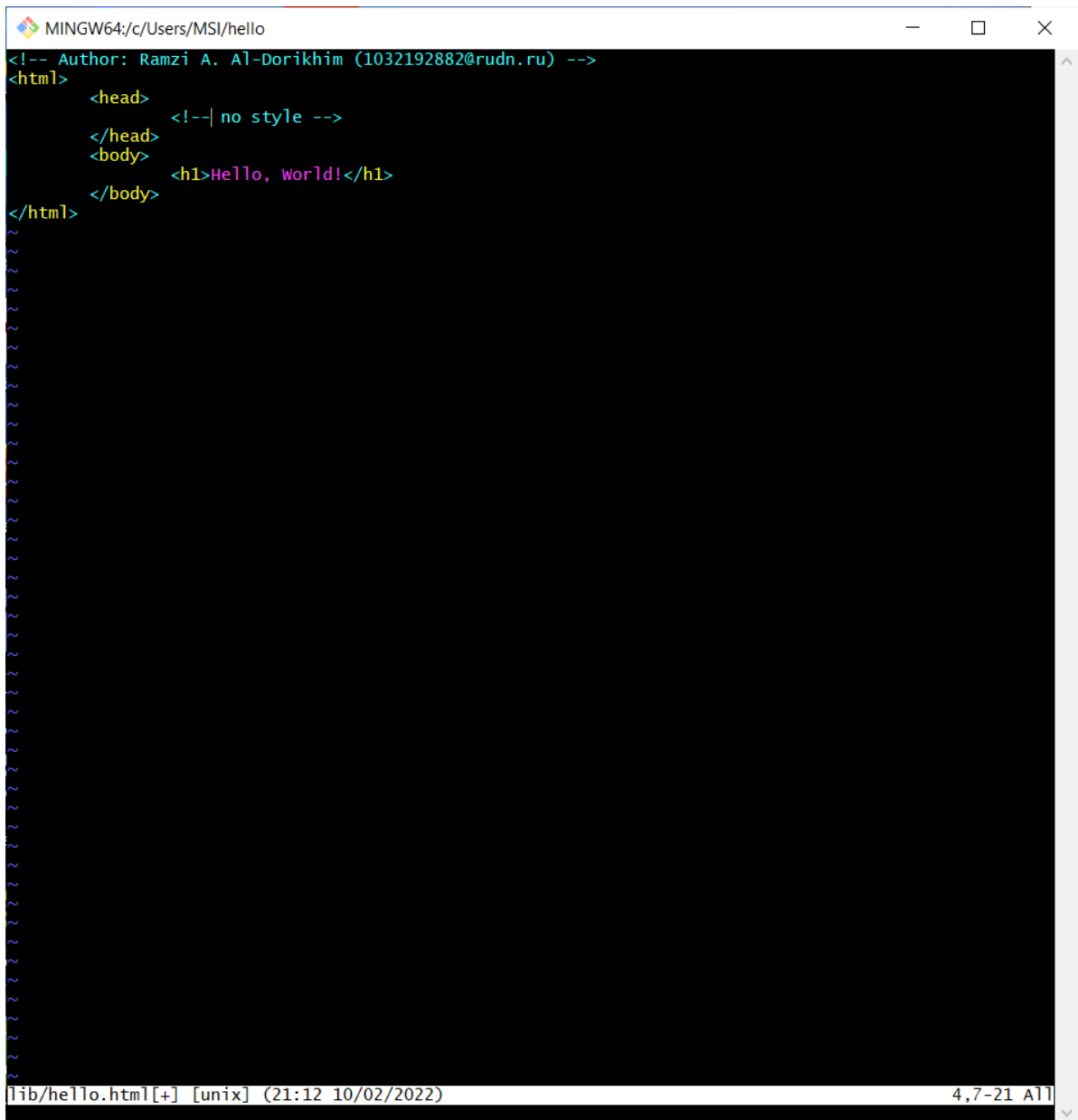
```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git log --graph --all
*   commit 1c2c2ef061157d0f1d2c96931a53ace515027968 (HEAD -> style)
|  \ Merge: a66018c 90c14ed
|   Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
|   Date: Thu Feb 10 21:10:26 2022 +0300
|
|   Merge branch 'master' into style
|
| *   commit 90c14eda09352974490b9ee1b65dcc15b4b82283 (master)
|  \ Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
|   Date: Thu Feb 10 21:07:05 2022 +0300
|
|   Added README
|
| *   commit a66018cbc9df8cbbef619369227687c5efe3884f
|  \ Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
|   Date: Thu Feb 10 21:02:26 2022 +0300
|
|   Added css stylesheet
|
| *   commit f8a546f8a5ee894d3cf8977bbb96b60132a1d8f8
|  \ Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
|   Date: Thu Feb 10 20:56:48 2022 +0300
|
|   Updated index.html
|
| *   commit 6ca361ed6da661d42b3a12d2261defd6e72ae707
|  \ Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
|   Date: Thu Feb 10 20:52:47 2022 +0300
|
|   Hello uses style.css
|
| *   commit 5e9b6314341f517533858ff558e317c48e2c77d7
|  \ Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
|   Date: Thu Feb 10 20:45:36 2022 +0300
|
|   Added css stylesheet
|
| *   commit ca32d321743babdce514ff01c366316ec39a3c08
|  \ Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
|   Date: Thu Feb 10 19:24:12 2022 +0300
|
|   Added index.html.
|
| *   commit 39a89573b6c21978a0349951d48eafe3b57de830
|  \ Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
|   Date: Thu Feb 10 19:14:28 2022 +0300
|
|   Moved hello.html to lib
|
| *   commit 0f2a3a0da23832e25fc425d918aadf809c9ac30c
|  \ Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
|   Date: Thu Feb 10 18:59:16 2022 +0300
```

Путем периодического слияния ветки master с веткой style вы можете переносить из master любые изменения и поддерживать совместимость изменений style с изменениями в основной ветке.

1.21 Создание конфликта

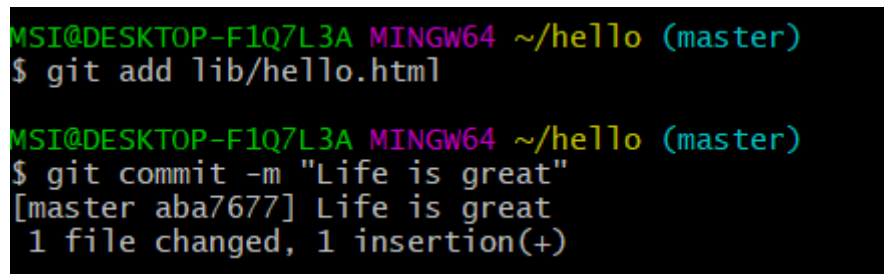
1.21.1 Вернитесь в master и создайте конфликт

Вернемся в ветку master и внесем следующие изменения. (рис. 087)



```
MINGW64:/c/Users/MSI/hello
<!-- Author: Ramzi A. Al-Dorikhim (1032192882@rudn.ru) -->
<html>
  <head>
    <!-- no style -->
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

Выполним следующие команды. (рис. 088)



```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git add lib/hello.html

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git commit -m "Life is great"
[master aba7677] Life is great
1 file changed, 1 insertion(+)
```

1.21.2 Просмотр веток

Выполним следующую команду. (рис. 089)

```
MINGW64: c:/Users/MSI/hello
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git log --graph --all
* commit aba76775e087cc416bfe5b45d3e5444954607532 (HEAD -> master)
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 21:14:03 2022 +0300

    Life is great

* commit 1c2c2ef061157d0f1d2c96931a53ace515027968 (style)
  Merge: a66018c 90c14ed
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 21:10:26 2022 +0300

    Merge branch 'master' into style

* commit 90c14eda09352974490b9ee1b65dcc15b4b82283
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 21:07:05 2022 +0300

    Added README

* commit a66018cbc9df8cbbef619369227687c5efe3884f
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 21:02:26 2022 +0300

    Added css stylesheet

* commit f8a546f8a5ee894d3cf8977bbb96b60132a1d8f8
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 20:56:48 2022 +0300

    Updated index.html

* commit 6ca361ed6da661d42b3a12d2261def6d6e72ae707
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 20:52:47 2022 +0300

    Hello uses style.css

* commit 5e9b6314341f517533858ff558e317c48e2c77d7
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 20:45:36 2022 +0300

    Added css stylesheet

* commit ca32d321743babdce514ff01c366316ec39a3c08
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 19:24:12 2022 +0300

    Added index.html.

* commit 39a89573b6c21978a0349951d48eafe3b57de830
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 19:14:28 2022 +0300
```

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. На следующем шаге мы решим этот конфликт.

1.22 Разрешение конфликтов

1.22.1 Слияние master с веткой style

Теперь вернемся к ветке style и попытаемся объединить ее с новой веткой master.

Выполним следующие команды. (рис. 090)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git checkout style
Switched to branch 'style'

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git merge master
Auto-merging lib/hello.html
CONFLICT (content): Merge conflict in lib/hello.html
Automatic merge failed; fix conflicts and then commit the result.
```

Откроем lib/hello.html. (рис. 091)


```

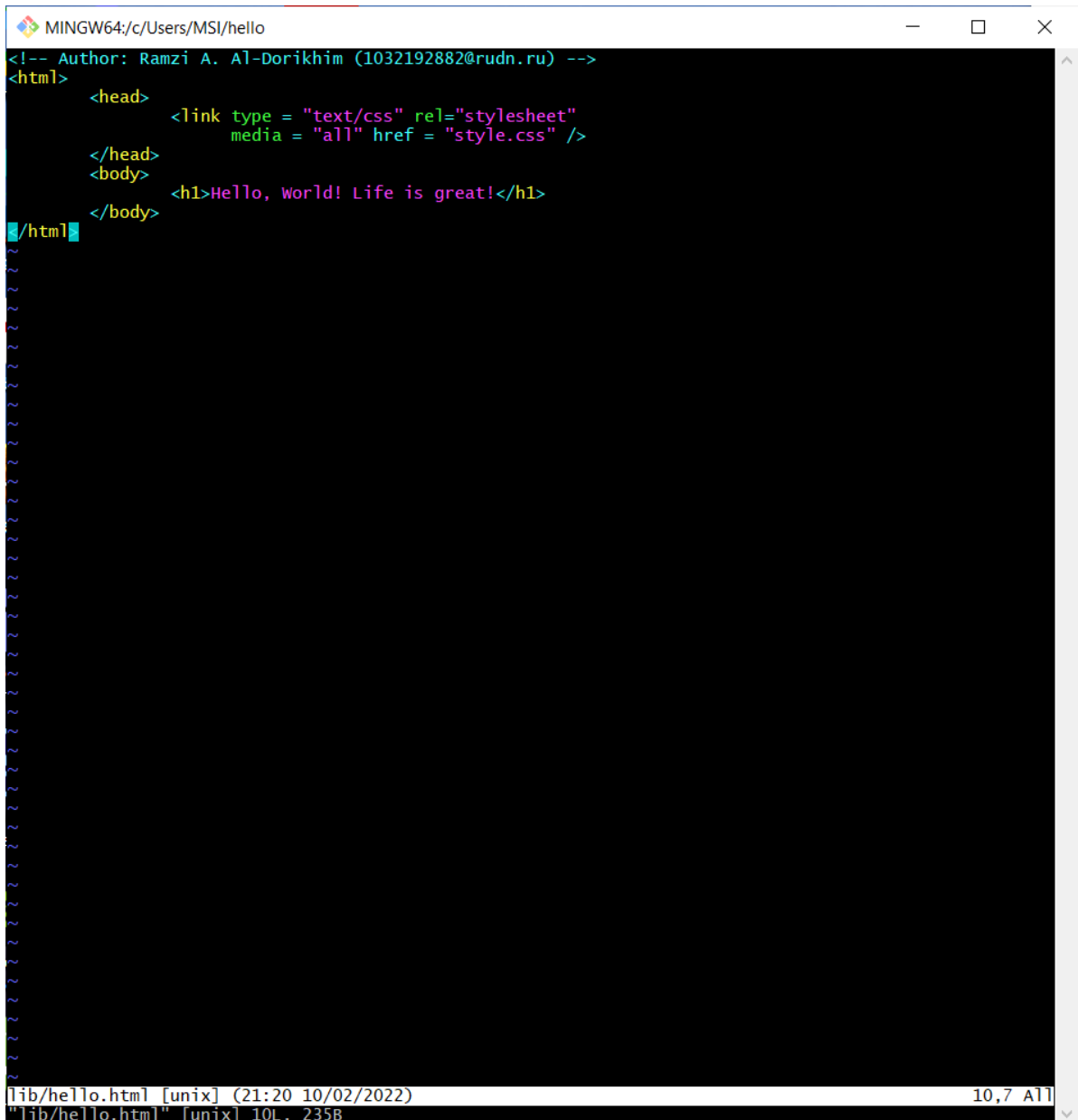
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style|MERGING)
$ cat lib/hello.html
<!-- Author: Ramzi A. Al-Dorikhim (1032192882@rudn.ru) -->
<html>
  <head>
<<<<<< HEAD
    <link type = "text/css" rel="stylesheet"
      media = "all" href = "style.css" />
=====
    <!-- no style -->
>>>>>> master
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>

```

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

1.22.2 Решение конфликта

Нам необходимо вручную разрешить конфликт. Внесем изменения в lib/hello.html для достижения следующего результата. (рис. 092)



```

MINGW64:/c/Users/MSI/hello
<!-- Author: Ramzi A. Al-Dorikhim (1032192882@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>

```

lib/hello.html [unix] (21:20 10/02/2022) 10,7 All
 "lib/hello.html" [unix] 10L, 235B

1.22.3 Сделаем коммит решения конфликта

Выполним следующие команды. (рис. 093)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style|MERGING)
$ git add lib/hello.html

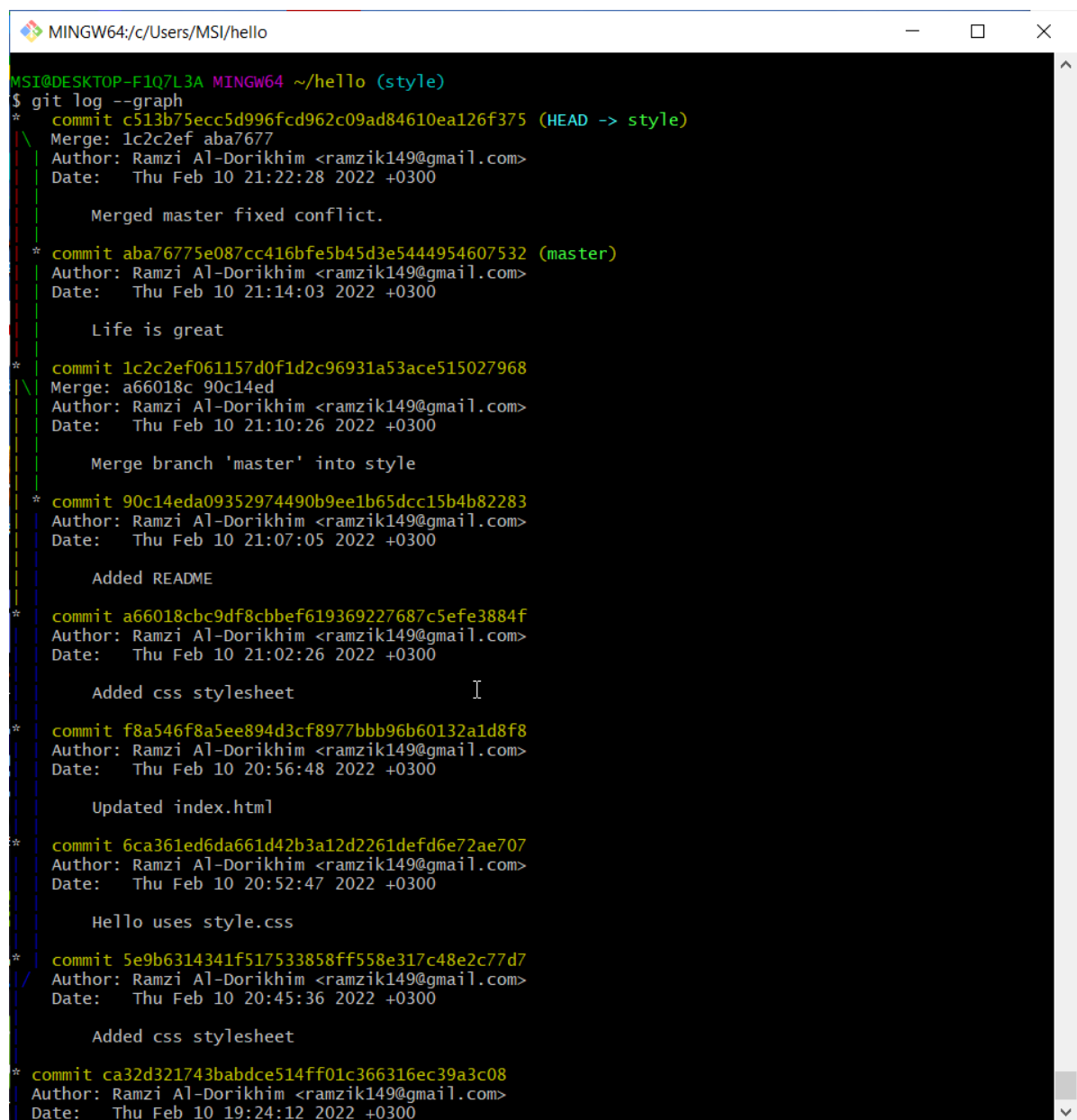
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style|MERGING)
$ git commit -m "Merged master fixed conflict."
[style c513b75] Merged master fixed conflict.
```

1.23 Сброс ветки style

1.23.1 Сброс ветки style

Вернемся на ветке style к точке перед тем, как мы слили ее с веткой master. Нам необходимо найти последний коммит перед слиянием.

Выполним следующие команды. (рис. 094)



```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git log --graph
*   commit c513b75ecc5d996fcd962c09ad84610ea126f375 (HEAD -> style)
  \  Merge: 1c2c2ef aba7677
   | Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
   | Date: Thu Feb 10 21:22:28 2022 +0300
   |
   | Merged master fixed conflict.
   |
*   commit aba76775e087cc416bfe5b45d3e5444954607532 (master)
  | Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  | Date: Thu Feb 10 21:14:03 2022 +0300
  |
  | Life is great
  |
*   commit 1c2c2ef061157d0f1d2c96931a53ace515027968
  \  Merge: a66018c 90c14ed
   | Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
   | Date: Thu Feb 10 21:10:26 2022 +0300
   |
   | Merge branch 'master' into style
   |
*   commit 90c14eda09352974490b9ee1b65dcc15b4b82283
  | Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  | Date: Thu Feb 10 21:07:05 2022 +0300
  |
  | Added README
  |
*   commit a66018cbc9df8cbbef619369227687c5efe3884f
  | Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  | Date: Thu Feb 10 21:02:26 2022 +0300
  |
  | Added css stylesheet
  |
*   commit f8a546f8a5ee894d3cf8977bbb96b60132a1d8f8
  | Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  | Date: Thu Feb 10 20:56:48 2022 +0300
  |
  | Updated index.html
  |
*   commit 6ca361ed6da661d42b3a12d2261defd6e72ae707
  | Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  | Date: Thu Feb 10 20:52:47 2022 +0300
  |
  | Hello uses style.css
  |
*   commit 5e9b6314341f517533858ff558e317c48e2c77d7
  | Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  | Date: Thu Feb 10 20:45:36 2022 +0300
  |
  | Added css stylesheet
  |
*   commit ca32d321743babdce514ff01c366316ec39a3c08
  | Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  | Date: Thu Feb 10 19:24:12 2022 +0300
```

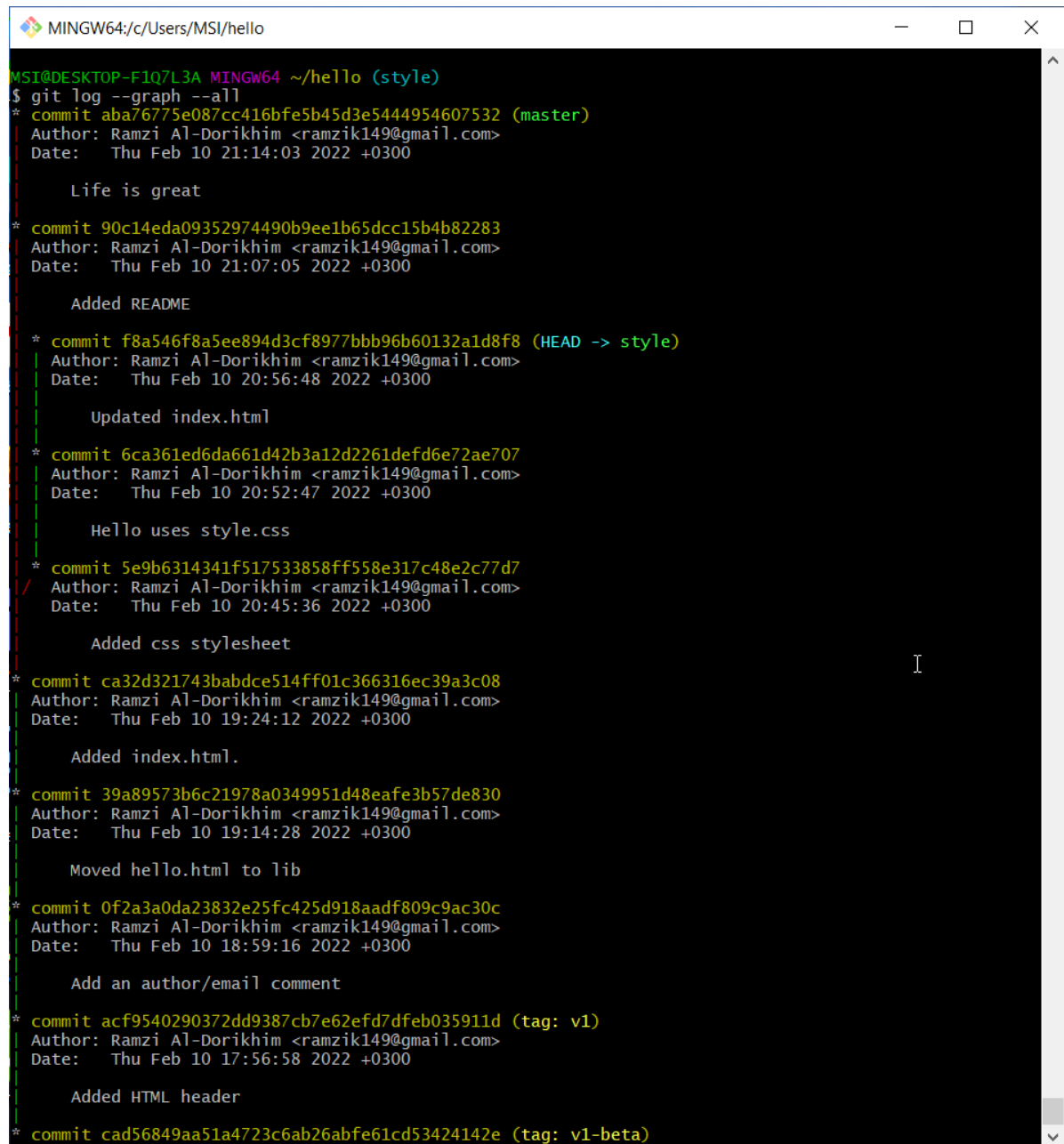
Мы видим, что коммит «Updated index.html» был последним на ветке style перед слиянием. Давайте сбросим ветку style к этому коммиту.

Выполним следующую команду. (рис. 095)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git reset --hard f8a546f8a5ee894d3cf8977bbb96b60132a1d8f8
HEAD is now at f8a546f Updated index.html
```

1.23.2 Проверьте ветку

Поищем лог ветки style. У нас в истории больше нет коммитов слияний. (рис. 096)



```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git log --graph --all
* commit aba76775e087cc416bfe5b45d3e5444954607532 (master)
| Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
| Date: Thu Feb 10 21:14:03 2022 +0300
|
| Life is great
|
* commit 90c14eda09352974490b9ee1b65dcc15b4b82283
| Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
| Date: Thu Feb 10 21:07:05 2022 +0300
|
| Added README
|
* commit f8a546f8a5ee894d3cf8977bbb96b60132a1d8f8 (HEAD -> style)
| Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
| Date: Thu Feb 10 20:56:48 2022 +0300
|
| Updated index.html
|
* commit 6ca361ed6da661d42b3a12d2261defd6e72ae707
| Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
| Date: Thu Feb 10 20:52:47 2022 +0300
|
| Hello uses style.css
|
* commit 5e9b6314341f517533858ff558e317c48e2c77d7
| Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
| Date: Thu Feb 10 20:45:36 2022 +0300
|
| Added css stylesheet
|
* commit ca32d321743babdce514ff01c366316ec39a3c08
| Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
| Date: Thu Feb 10 19:24:12 2022 +0300
|
| Added index.html.
|
* commit 39a89573b6c21978a0349951d48eafe3b57de830
| Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
| Date: Thu Feb 10 19:14:28 2022 +0300
|
| Moved hello.html to lib
|
* commit 0f2a3a0da23832e25fc425d918aadf809c9ac30c
| Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
| Date: Thu Feb 10 18:59:16 2022 +0300
|
| Add an author/email comment
|
* commit acf9540290372dd9387cb7e62efd7dfef035911d (tag: v1)
| Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
| Date: Thu Feb 10 17:56:58 2022 +0300
|
| Added HTML header
|
* commit cad56849aa51a4723c6ab26abfe61cd53424142e (tag: v1-beta)
```

1.24 Сброс ветки master

1.24.1 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.

Выполним следующие команды. (рис. 097)

```
MINGW64:/c/Users/MSI/hello

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git checkout master
Switched to branch 'master'

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git log --graph
* commit aba76775e087cc416bfe5b45d3e5444954607532 (HEAD -> master)
| Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
| Date: Thu Feb 10 21:14:03 2022 +0300
|
| Life is great
|
* commit 90c14eda09352974490b9ee1b65dcc15b4b82283
| Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
| Date: Thu Feb 10 21:07:05 2022 +0300
|
| Added README
|
* commit ca32d321743babdce514ff01c366316ec39a3c08
| Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
| Date: Thu Feb 10 19:24:12 2022 +0300
|
| Added index.html.
|
* commit 39a89573b6c21978a0349951d48eafe3b57de830
| Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
| Date: Thu Feb 10 19:14:28 2022 +0300
|
| Moved hello.html to lib
|
* commit 0f2a3a0da23832e25fc425d918aadf809c9ac30c
| Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
| Date: Thu Feb 10 18:59:16 2022 +0300
|
| Add an author/email comment
|
* commit acf9540290372dd9387cb7e62efd7dfef035911d (tag: v1)
| Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
| Date: Thu Feb 10 17:56:58 2022 +0300
|
| Added HTML header
|
* commit cad56849aa51a4723c6ab26abfe61cd53424142e (tag: v1-beta)
| Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
| Date: Thu Feb 10 17:53:05 2022 +0300
|
| Added standart HTML page tags
|
* commit 5ac1631fe79849dff360aac7d21ebcf05c3c1245
| Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
| Date: Thu Feb 10 17:38:25 2022 +0300
|
| Added h1 tag
|
* commit 6840ea78eecac12d3180226f0801d52ad7c0eb5a
| Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
| Date: Thu Feb 10 17:20:21 2022 +0300
```

Коммит «Added README» идет непосредственно перед коммитом конфликтующего интерактивного режима. Мы сбросим ветку master к коммиту «Added README».

Выполним следующие команды. (рис. 098)(рис. 099)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git reset --hard 90c14eda09352974490b9ee1b65dcc15b4b82283
HEAD is now at 90c14ed Added README
```

```
MINGW64/c/Users/MSI/hello
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git log --graph --all
* commit 90c14eda09352974490b9ee1b65dcc15b4b82283 (HEAD -> master)
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 21:07:05 2022 +0300

    Added README

* commit f8a546f8a5ee894d3cf8977bbb96b60132a1d8f8 (style)
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 20:56:48 2022 +0300

    Updated index.html

* commit 6ca361ed6da661d42b3a12d2261defd6e72ae707
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 20:52:47 2022 +0300

    Hello uses style.css

* commit 5e9b6314341f517533858ff558e317c48e2c77d7
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 20:45:36 2022 +0300

    Added css stylesheet

* commit ca32d321743babdce514ff01c366316ec39a3c08
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 19:24:12 2022 +0300

    Added index.html.

* commit 39a89573b6c21978a0349951d48eafe3b57de830
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 19:14:28 2022 +0300

    Moved hello.html to lib

* commit 0f2a3a0da23832e25fc425d918aadf809c9ac30c
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 18:59:16 2022 +0300

    Add an author/email comment

* commit acf9540290372dd9387cb7e62efd7dfef035911d (tag: v1)
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 17:56:58 2022 +0300

    Added HTML header

* commit cad56849aa51a4723c6ab26abfe61cd53424142e (tag: v1-beta)
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 17:53:05 2022 +0300

    Added standart HTML page tags

* commit 5ac1631fe79849dff360aac7d21ebcf05c3c1245
```

Мы видим, что репозиторий был будто перемотан назад во времени.

1.25 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.

Выполним следующие команды. (рис. 100) (рис. 101)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git checkout style
Switched to branch 'style'

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git rebase master
Successfully rebased and updated refs/heads/style.
```

```
MINGW64:/c/Users/MSI/hello
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git log --graph
* commit 231d42f1221142c16caa85504d7b06c4b2425c18 (HEAD -> style)
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 20:56:48 2022 +0300

    Updated index.html

* commit 2f64f6a45f73ba2f4dfc5e8bc27c34f715caaf25
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 20:52:47 2022 +0300

    Hello uses style.css

* commit 58d88d8129f781acc7d4ee8a44328835265f436a
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 20:45:36 2022 +0300

    Added css stylesheet

* commit 90c14eda09352974490b9ee1b65dcc15b4b82283 (master)
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 21:07:05 2022 +0300

    Added README

* commit ca32d321743babdce514ff01c366316ec39a3c08
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 19:24:12 2022 +0300

    Added index.html.

* commit 39a89573b6c21978a0349951d48eafe3b57de830
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 19:14:28 2022 +0300

    Moved hello.html to lib

* commit 0f2a3a0da23832e25fc425d918aadf809c9ac30c
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 18:59:16 2022 +0300

    Add an author/email comment

* commit acf9540290372dd9387cb7e62efd7dfeb035911d (tag: v1)
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 17:56:58 2022 +0300

    Added HTML header

* commit cad56849aa51a4723c6ab26abfe61cd53424142e (tag: v1-beta)
  Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
  Date: Thu Feb 10 17:53:05 2022 +0300

    Added standart HTML page tags

* commit 5ac1631fe79849dff360aac7d21ebcf05c3c1245
```

1.25.1 Слияние VS перебазирование

Конечный результат перебазирования очень похож на результат слияния. Ветка style в настоящее время содержит все свои изменения, а также все изменения ветки master. Однако, дерево коммитов значительно отличается. Дерево коммитов ветки style было переписано таким образом, что ветка master является частью истории коммитов. Это делает цепь коммитов линейной и гораздо более читабельной.

1.26 Слияние в ветку master

Мы поддерживали соответствие ветки style с веткой master (с помощью rebase), теперь давайте сольем изменения style в ветку master.

1.26.1 Слияние style в master

Выполним следующие команды. (рис. 102)

```

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (style)
$ git checkout master
Switched to branch 'master'

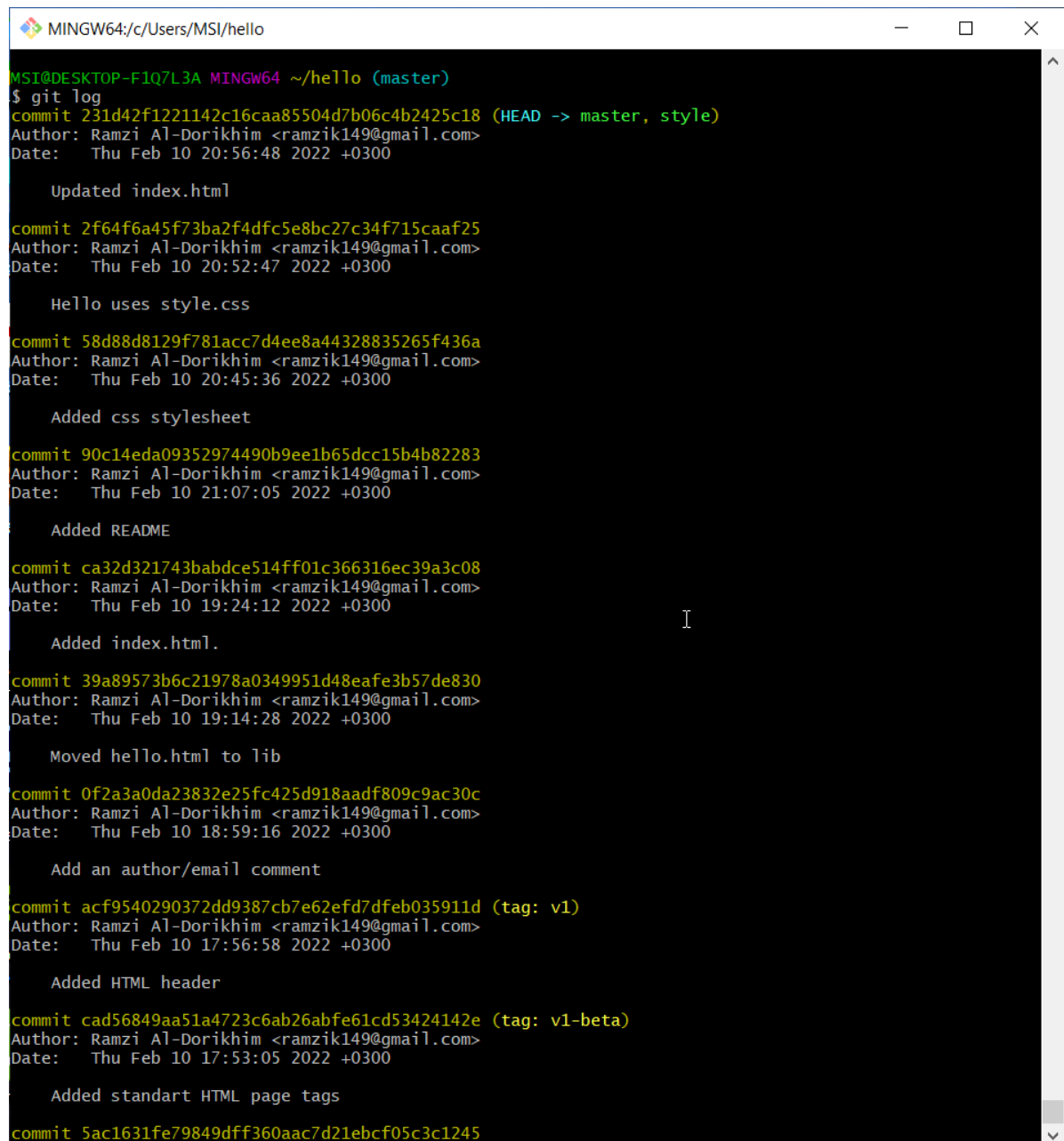
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git merge style
Updating 90c14ed..231d42f
Fast-forward
 index.html      | 4 ++++
 lib/hello.html | 2 ++
 lib/style.css   | 0
 3 files changed, 6 insertions(+)
 create mode 100644 lib/style.css

```

Поскольку последний коммит ветки master прямо предшествует последнему коммиту ветки style, git может выполнить ускоренное слияние-перемотку. При быстрой перемотке вперед git просто передвигает указатель вперед, таким образом указывая на тот же коммит, что и ветка style. При быстрой перемотке конфликтов быть не может.

1.26.2 Просмотрите логи

Выполним следующую команду. (рис. 103)



```

MINGW64: c:/Users/MSI/hello

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git log
commit 231d42f1221142c16caa85504d7b06c4b2425c18 (HEAD -> master, style)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 20:56:48 2022 +0300

    Updated index.html

commit 2f64f6a45f73ba2f4dfc5e8bc27c34f715caaf25
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 20:52:47 2022 +0300

    Hello uses style.css

commit 58d88d8129f781acc7d4ee8a44328835265f436a
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 20:45:36 2022 +0300

    Added css stylesheet

commit 90c14eda09352974490b9ee1b65dcc15b4b82283
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 21:07:05 2022 +0300

    Added README

commit ca32d321743babdce514ff01c366316ec39a3c08
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 19:24:12 2022 +0300

    Added index.html.

commit 39a89573b6c21978a0349951d48eafe3b57de830
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 19:14:28 2022 +0300

    Moved hello.html to lib

commit 0f2a3a0da23832e25fc425d918aadf809c9ac30c
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 18:59:16 2022 +0300

    Add an author/email comment

commit acf9540290372dd9387cb7e62efd7dfef035911d (tag: v1)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:56:58 2022 +0300

    Added HTML header

commit cad56849aa51a4723c6ab26abfe61cd53424142e (tag: v1-beta)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:53:05 2022 +0300

    Added standart HTML page tags

commit 5ac1631fe79849dff360aac7d21ebcf05c3c1245

```

Теперь ветки style и master идентичны.

1.27 Клонирование репозиториев

1.27.1 Перейдите в рабочий каталог

Перейдем в рабочий каталог и сделаем клон вашего репозитория hello.

Выполним следующие команды. (рис. 104)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ cd

MSI@DESKTOP-F1Q7L3A MINGW64 ~
$ pwd
/c/Users/MSI

MSI@DESKTOP-F1Q7L3A MINGW64 ~
$ ls
02.09.2021_Lab1.ipynb
'03.09.2021- Methods.ipynb'
09.09.2021_AI.ipynb
'3D objects'/'
7/
```

1.27.2 Создадим клон репозитория hello

Выполним следующие команды:

```
git clone hello cloned_hello
```

```
ls
```

(рис. 105)

```
cloned_hello/
content-collaborative-anime-recommendation.ipynb
dataset_209691_15.txt
dataset_209770_6.txt
db
db_sber
db_test
firstspell.ipynb
hello/
```

Мы видим в рабочем каталоге теперь два репозитория: оригинальный репозиторий «hello» и клонированный репозиторий «cloned_hello»

1.28 Просмотр клонированного репозитория

1.28.1 Давайте взглянем на клонированный репозиторий.

Выполним следующие команды. (рис. 106)

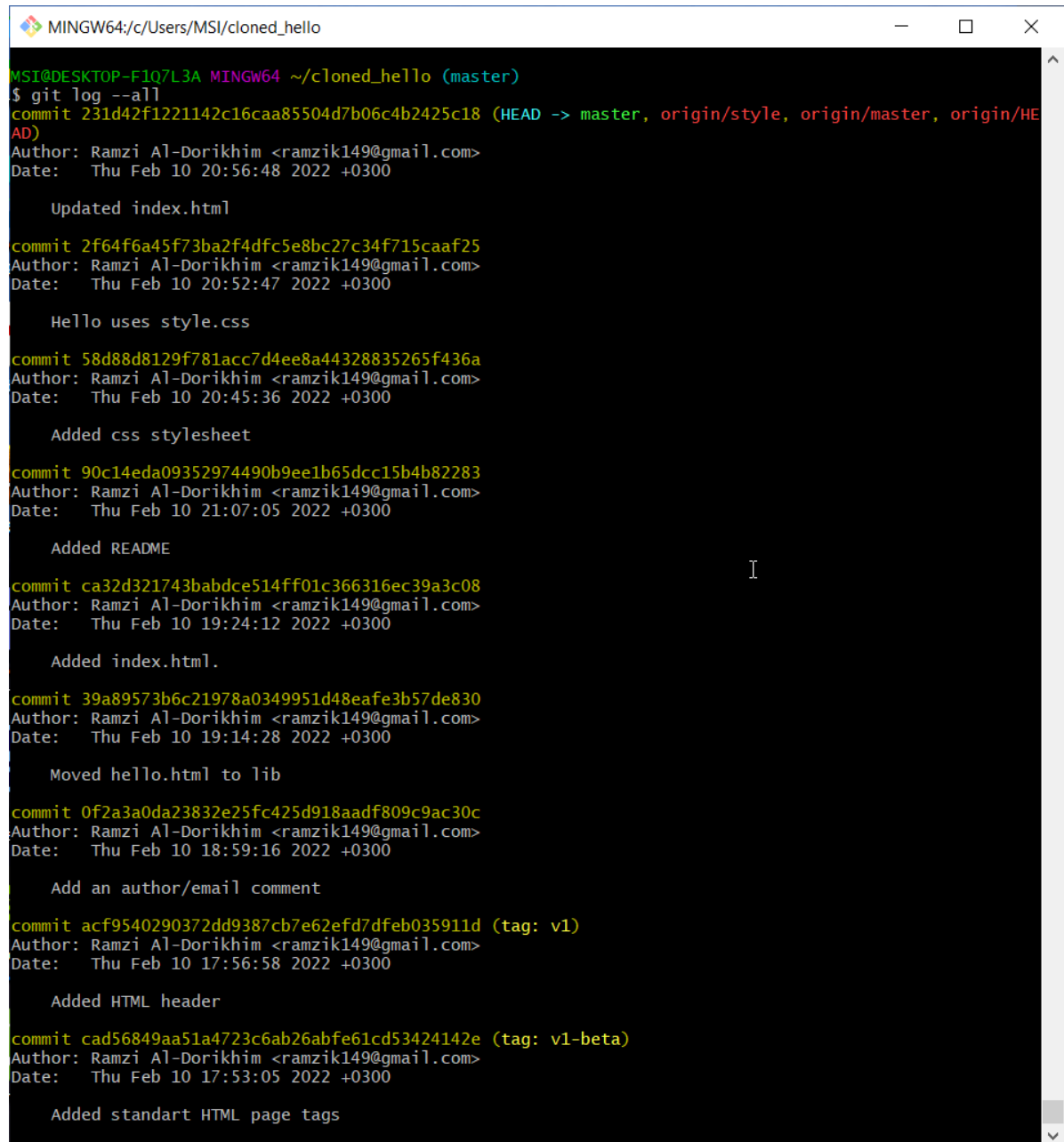
```
MSI@DESKTOP-F1Q7L3A MINGW64 ~
$ cd cloned_hello

MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ ls
README.md  index.html  lib/
```


Мы увидим список всех файлов на верхнем уровне оригинального репозитория README.md, index.html и lib.

1.28.2 Просмотрите историю репозитория

Выполним следующую команду. (рис. 107)



```
MINGW64: c:/Users/MSI/cloned_hello
MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ git log --all
commit 231d42f1221142c16caa85504d7b06c4b2425c18 (HEAD -> master, origin/style, origin/master, origin/HEAD)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 20:56:48 2022 +0300

    Updated index.html

commit 2f64f6a45f73ba2f4dfc5e8bc27c34f715caaf25
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 20:52:47 2022 +0300

    Hello uses style.css

commit 58d88d8129f781acc7d4ee8a44328835265f436a
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 20:45:36 2022 +0300

    Added css stylesheet

commit 90c14eda09352974490b9ee1b65dcc15b4b82283
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 21:07:05 2022 +0300

    Added README

commit ca32d321743babdce514ff01c366316ec39a3c08
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 19:24:12 2022 +0300

    Added index.html.

commit 39a89573b6c21978a0349951d48eafe3b57de830
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 19:14:28 2022 +0300

    Moved hello.html to lib

commit 0f2a3a0da23832e25fc425d918aadf809c9ac30c
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 18:59:16 2022 +0300

    Add an author/email comment

commit acf9540290372dd9387cb7e62efd7dfeb035911d (tag: v1)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:56:58 2022 +0300

    Added HTML header

commit cad56849aa51a4723c6ab26abfe61cd53424142e (tag: v1-beta)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:53:05 2022 +0300

    Added standart HTML page tags
```

Мы увидели список всех коммитов в новый репозиторий, и он должен (более или менее) совпадать с историей коммитов в оригинальном репозитории. Также здесь мы увидим ветки со странными именами (origin/master, origin/style и origin/HEAD).

1.29 Что такое origin?

Выполним следующие команды. (рис. 108)

```

MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ git remote
origin

MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ git remote show origin
* remote origin
Fetch URL: C:/Users/MSI/hello
Push URL: C:/Users/MSI/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)

```

Выполнив `git remote`, Мы видим, что клонированный репозиторий знает об имени по умолчанию удаленного репозитория. Давайте посмотрим, можем ли мы получить более подробную информацию об имени по умолчанию.

Выполнив `git remote show origin`, мы увидели что удаленные репозитории обычно размещаются на отдельной машине, возможно, централизованном сервере. Однако, как мы видим здесь, они могут с тем же успехом указывать на репозиторий на той же машине. Нет ничего особенного в имени «origin», однако существует традиция использовать «origin» в качестве имени первичного централизованного репозитория (если таковой имеется).

1.30 Удаленные ветки

Давайте посмотрим на ветки, доступные в нашем клонированном репозитории.

Выполним следующую команду. (рис. 109)

```

MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ git branch
* master

```

Как мы видим, в списке только ветка `master`. Команда `git branch` выводит только список локальных веток по умолчанию.

1.30.1 Список удаленных веток

Для того, чтобы увидеть все ветки, попробуем следующую команду. (рис. 110)

```

MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ git branch -a
* master
remotes/origin/HEAD -> origin/master
remotes/origin/master
remotes/origin/style

```

Git выводит все коммиты в оригинальный репозиторий, но ветки в удаленном репозитории не рассматриваются как локальные.

1.31 Изменение оригинального репозитория

Внесем некоторые изменения в оригинальный репозиторий, чтобы затем попытаться извлечь и слить изменения из удаленной ветки в текущую.

1.31.1 Внесите изменения в оригинальный репозиторий hello

Выполним следующую команду и отредактируем файл README.md. (рис. 111)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ cd ../hello

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ vi README.md
```

Теперь добавим это изменение и сделаем коммит. (рис. 112)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git add README.md

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git commit -m "Changed README in original repo"
[master 369a8e6] Changed README in original repo
1 file changed, 1 insertion(+), 1 deletion(-)
```

Слово World я заменил на world.

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

1.31.2 Извлечение изменений

Научиться извлекать изменения из удаленного репозитория.

Выполним следующие команды. (рис. 113) (рис. 114)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ cd ../cloned_hello

MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 362 bytes | 32.00 KiB/s, done.
From C:/Users/MSI/hello
 231d42f..369a8e6  master      -> origin/master
```

```
MINGW64/c/Users/MSI/cloned_hello
MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ git log --all
commit 369a8e6ed92e0eed87e122a9b84cf8c7c25db0a2 (origin/master, origin/HEAD)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 21:57:29 2022 +0300

    Changed README in original repo

commit 231d42f1221142c16caa85504d7b06c4b2425c18 (HEAD -> master, origin/style)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 20:56:48 2022 +0300

    Updated index.html

commit 2f64f6a45f73ba2f4dfc5e8bc27c34f715caaf25
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 20:52:47 2022 +0300

    Hello uses style.css

commit 58d88d8129f781acc7d4ee8a44328835265f436a
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 20:45:36 2022 +0300

    Added css stylesheet

commit 90c14eda09352974490b9ee1b65dcc15b4b82283
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 21:07:05 2022 +0300

    Added README

commit ca32d321743babdce514ff01c366316ec39a3c08
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 19:24:12 2022 +0300

    Added index.html.

commit 39a89573b6c21978a0349951d48eafe3b57de830
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 19:14:28 2022 +0300

    Moved hello.html to lib

commit 0f2a3a0da23832e25fc425d918aadf809c9ac30c
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 18:59:16 2022 +0300

    Add an author/email comment

commit acf9540290372dd9387cb7e62efd7dfef035911d (tag: v1)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 17:56:58 2022 +0300

    Added HTML header

commit cad56849aa51a4723c6ab26abfe61cd53424142e (tag: v1-beta)
```

Сейчас мы находимся в репозитории cloned_hello. На данный момент в репозитории есть все коммиты из оригинального репозитория, но они не интегрированы в локальные ветки клонированного репозитория.

1.31.3 Проверим README.md

Мы можем продемонстрировать, что клонированный файл README.md не изменился. (рис. 115)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ cat README.md
This is the Hello World example from the git tutorial.
```

1.32 Слияние извлеченных изменений

1.32.1 Слейте извлеченные изменения в локальную ветку master

Выполним.(рис. 116)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ git merge origin/master
Updating 231d42f..369a8e6
Fast-forward
 README.md | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

1.32.2 Еще раз проверьте файл README.md

Сейчас мы должны увидеть изменения.

Выполним.(рис. 117)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ cat README.md
This is the Hello world example from the git tutorial.
```

Отличие между 115 и 117 рисунками в слове World. Сначала с большой буквы, а потом с маленькой.

Теперь давайте рассмотрим объединение fetch и merge в одну команду. (рис. 118)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ git pull
Already up to date.
```

Выполнение git pull эквивалентно двум следующим шагам:

git fetch

git merge origin/master

1.33 Добавление ветки наблюдения

Ветки, которые начинаются с remotes/origin являются ветками оригинального репозитория. Обратим внимание, что у вас больше нет ветки под названием style, но система контроля версий знает, что в оригинальном репозитории ветка style была.

1.33.1 Добавим локальную ветку, которая отслеживает удаленную ветку

Выполним. (рис. 119)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ git branch --track style origin/style
branch 'style' set up to track 'origin/style'.

MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ git branch -a
* master
  style
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/style

MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ git log --max-count=2
commit 369a8e6ed92e0eed87e122a9b84cf8c7c25db0a2 (HEAD -> master, origin/master, origin/HEAD)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 21:57:29 2022 +0300

    Changed README in original repo

commit 231d42f1221142c16caa85504d7b06c4b2425c18 (origin/style, style)
Author: Ramzi Al-Dorikhim <ramzik149@gmail.com>
Date: Thu Feb 10 20:56:48 2022 +0300

    Updated index.html
```

Теперь мы можем видеть ветку style в списке веток и логе.

1.34 Чистые репозитории

Чистые репозитории (без рабочих каталогов) обычно используются для расшаривания. Обычный git-репозиторий подразумевает, что вы будете использовать его как рабочую директорию, поэтому вместе с файлами проекта в актуальной версии, git хранит все служебные, «чисто-репозиториевские» файлы в поддиректории .git. В удаленных репозиториях нет смысла хранить рабочие файлы на диске (как это делается в рабочих копиях), а все что им действительно нужно — это дельты изменений и другие бинарные данные репозитория. Вот это и есть «чистый репозиторий».

1.35 Создаем чистый репозиторий

Выполним. (рис. 120)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ cd ..

MSI@DESKTOP-F1Q7L3A MINGW64 ~
$ git clone --bare hello hello.git
Cloning into bare repository 'hello.git'...
done.

MSI@DESKTOP-F1Q7L3A MINGW64 ~
$ ls hello.git
HEAD config description hooks/ info/ objects/ packed-refs refs/
```

Сейчас мы находимся в рабочем каталоге. Как правило, репозитории, оканчивающиеся на .git являются чистыми репозиториями. Мы видим, что в репозитории hello.git нет рабочего каталога. По сути, это есть не что иное, как каталог .git нечистого репозитория.

1.36 Добавление удаленного репозитория

Давайте добавим репозиторий hello.git к нашему оригинальному репозиторию. (рис. 121)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~
$ cd hello

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git remote add shared ../hello.git
```

1.37 Отправка изменений

Так как чистые репозитории, как правило, расшариваются на каком-нибудь сетевом сервере, нам необходимо отправить наши изменения в другие репозитории. Начнем с создания изменения для отправки. Отредактируем файл README.md и сделаем коммит. (рис. 122)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git add README.md

MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git commit -m "Added shared comment to readme"
[master b8e756e] Added shared comment to readme
1 file changed, 2 insertions(+), 1 deletion(-)
```

Теперь отправим изменения в общий репозиторий. (рис. 123)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ git push shared master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 412 bytes | 412.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To ../hello.git
    369a8e6..b8e756e  master -> master
```

1.38 Извлечение общих изменений

Научимся извлекать изменения из общего репозитория. (рис. 124)

```
MSI@DESKTOP-F1Q7L3A MINGW64 ~/hello (master)
$ cd ../cloned_hello

MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ git remote add shared ../hello.git

MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ git branch --track shared master
branch 'shared' set up to track 'master'.

MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ 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), 392 bytes | 30.00 KiB/s, done.
From ../hello
 * branch          master      -> FETCH_HEAD
 * [new branch]    master      -> shared/master
Updating 369a8e6..b8e756e
Fast-forward
 README.md | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)

MSI@DESKTOP-F1Q7L3A MINGW64 ~/cloned_hello (master)
$ cat README.md
This is the Hello World example from the git tutorial.
(Changed in the original and pushed to shared)
```

Вывод

Благодаря этой лабораторной работе я наконец-то пришел к Git Bash. Научился им пользоваться, что несомненно является очень полезным навыком.

Список литературы

Кулябов Д. С. *Лабораторная работа №1***: git.pdf

