ЛАБОРАТОРНАЯ РАБОТА №5

Создание конфликта

При слиянии двух веток Git пытается перенести изменения из одной ветки в другую. Если в обеих ветках была изменена одна и та же часть файла, Git может не справиться с автоматическим слиянием изменений. В этом случае Git сообщит о конфликте и попросит разрешить его вручную. В этом уроке мы смоделируем конфликт, а затем научимся его разрешать.

В реальной жизни конфликты слияния регулярно возникают при работе в команде. Например, вы и ваш коллега начали работать над двумя разными фичами, затрагивающими одни и те же файлы. Ваш коллега закончил работу первым и слил свои изменения в ветку main. Теперь вы хотите слить свои изменения в ветку main. Но ветка main теперь отличается от той, с которой вы начинали работать в начале — в ней появился новый код, присланный вашим коллегой. Вероятно, Git не сможет автоматически объединить ваши изменения и попросит помощи человека.

Вернитесь в main **и создайте конфликт**

Помните, что в нашей ветке main страница по-прежнему называется hello.html? Переключитесь обратно на ветку main и внесите следующие изменения:

```
git switch main
```

Файл: hello.html

```
<html>
  <head>
    <title>Hello World Page</title>
  </head>
  <body>
    <h1>Hello, World!</h1>
    <p>Let's learn Git together.</p>
  </body>
</html>
```

Просмотр веток

Выполните

```
git add hello.html
git commit -m "Added meta title"
```

Результат

```
$ git log --all --graph
* 85c14e9 2023-11-28 | Added meta title (HEAD -> main) [Alexander Shvets]
| *   a33deed 2023-11-28 | Merge branch 'main' into style (style) [Alexander Shvets]
| |\
| |/
|/|
* | ee16740 2023-11-28 | Added README [Alexander Shvets]
| * 0ee0113 2023-11-28 | Renamed hello.html; moved style.css [Alexander Shvets]
| * 903eb1d 2023-11-28 | Included stylesheet into hello.html [Alexander Shvets]
| * 555372e 2023-11-28 | Added css stylesheet [Alexander Shvets]
|/
* 9288a33 2023-11-28 | Added copyright statement with email [Alexander Shvets]
```

После коммита «Added README» ветка main была объединена с веткой style, но в настоящее время в main есть дополнительный коммит, который не был слит с style. Последнее изменение в main конфликтует с некоторыми изменениями в style. На следующем шаге мы решим этот конфликт.

Разрешение конфликтов

Слияние main **в ветку** style

Давайте вернемся в ветку style и сольем в нее все последние изменения из ветки main.

Выполните

```
git switch style
git merge main
```

Результат

```
$ git switch style
Switched to branch 'style'
$ git merge main
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

Похоже, что у нас возник конфликт. Ничего удивительного! Посмотрим, что скажет по этому поводу Git:

Выполните

```
git status
```

Результат

```
$ git status
On branch style
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   index.html

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

Если открыть файл `index.html`, то можно увидеть:

Файл: index.html

```
<!-- Author: Alexander Shvets (alex@githowto.com) -->
<html>
  <head>
<<<<<<< HEAD:index.html
    <link type="text/css" rel="stylesheet" media="all" href="style.css" />
=======
    <title>Hello World Page</title>
>>>>>>> main:hello.html
  </head>
  <body>
    <h1>Hello, World!</h1>
    <p>Let's learn Git together.</p>
  </body>
</html>
```

Часть между <<<<<<< >>>>>>> является конфликтом. Верхняя часть соответствует ветке style, которая является текущей веткой (или HEAD) репозитория. Нижняя часть соответствует изменениям из ветки main. Git не может решить, какие изменения применить, поэтому он просит вас разрешить конфликт вручную. Вы можете оставить изменения из ветки style или из main, либо объединить их любым удобным способом. Вы также можете внести в файл любые другие изменения.

Кстати, вы заметили, что наше второе изменение, тег <p>, не является частью конфликта? Это потому, что Git сумел автоматически объединить ее.

Отмена слияния

Прежде чем мы приступим к разрешению нашего конфликта, хочу заметить, что сразу бросаться к разрешению конфликта не всегда оптимально. Конфликт может быть вызван изменениями, о которых вы не знаете. Или же изменения слишком велики, чтобы разрешить конфликт сразу. По этой причине Git позволяет прервать слияние и вернуться к предыдущему состоянию. Для этого можно воспользоваться командой git merge --abort, как это было предложено командой status, которую мы выполнили ранее.

Выполните

```
git merge --abort
git status
```

Результат

```
$ git merge --abort
$ git status
On branch style
nothing to commit, working tree clean
```

Решение конфликта

После медитации мы готовы к разрешению конфликта ☺ Давайте снова запустим объединение.

Выполните

```
git merge main
```

Чтобы разрешить конфликт, нужно отредактировать файл до состояния, которое нас устраивает, и затем закоммитить его как обычно. В нашем случае мы объединим изменения из обеих веток. Поэтому мы отредактируем файл до следующего состояния:

Файл: index.html

```
<html>
  <head>
    <title>Hello World Page</title>
    <link type="text/css" rel="stylesheet" media="all" href="style.css" />
  </head>
  <body>
    <h1>Hello, World!</h1>
    <p>Let's learn Git together.</p>
  </body>
</html>
```

Закоммитьте разрешенный конфликт

Выполните

```
git add index.html
git commit -m "Resolved merge conflict"
git status
git log --all --graph
```

Результат

```
$ git add index.html
$ git commit -m "Resolved merge conflict"
[style 79ac6fa] Resolved merge conflict
```

Давайте посмотрим на текущее состояние нашего хранилища и убедимся, что все в порядке:

Выполните

```
git status
git log --all --graph
```

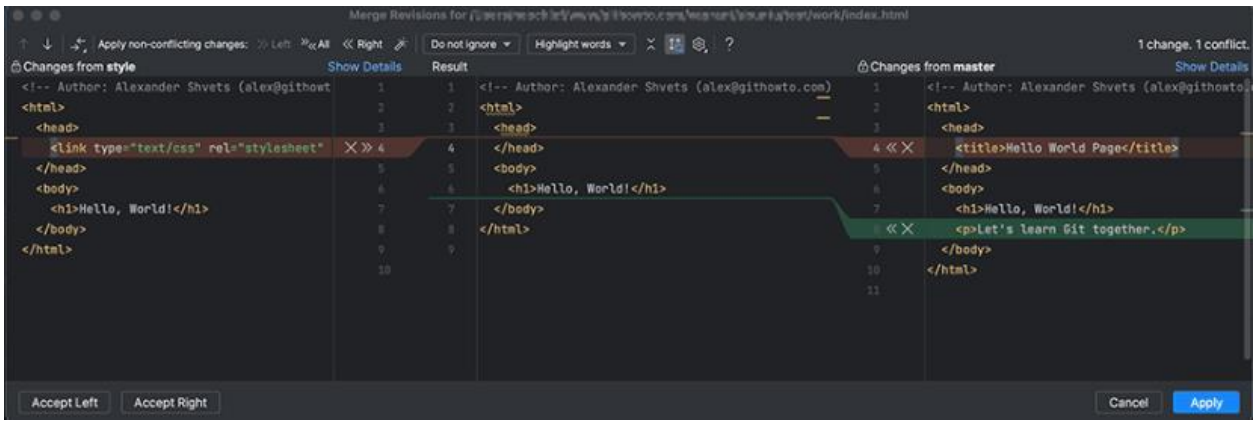
Результат

```
$ git status
On branch style
nothing to commit, working tree clean
$ git log --all --graph
*   79ac6fa 2023-11-28 | Resolved merge conflict (HEAD -> style) [Alexander Shvets]
|\
| * 85c14e9 2023-11-28 | Added meta title (main) [Alexander Shvets]
* | a33deed 2023-11-28 | Merge branch 'main' into style [Alexander Shvets]
|\|
| * ee16740 2023-11-28 | Added README [Alexander Shvets]
* | 0ee0113 2023-11-28 | Renamed hello.html; moved style.css [Alexander Shvets]
```

Расширенные возможности слияния

Git не предоставляет никаких графических инструментов слияния, но будет с удовольствием работать с любыми сторонними инструментами слияния, которые вы хотите использовать (обсуждение таких инструментов на StackOverflow).

При работе с кодом в современной интегрированной среде разработки (IDE), такой как Visual Studio или IntelliJ IDEA, вы, скорее всего, будете использовать встроенный инструмент слияния. Например, вот как выглядит разрешение конфликтов в IntelliJ IDEA:



rebase **против** merge

Обсуждение

Давайте рассмотрим различия между слиянием и перебазированием. Для того чтобы это сделать, нам нужно вернуться в репозиторий в момент до первого слияния, а затем повторить те же действия, но с использованием перебазирования вместо слияния.

Мы будем использовать команду reset для возврата веток к предыдущему состоянию.

Сброс ветки style

Сбросьте ветку style

Давайте вернемся во времени на ветке style к точке *перед* тем, как мы слили ее с веткой main. Мы можем сбросить ветку к любому коммиту при помощи команды reset. По сути, это изменение указателя ветки на любую точку дерева коммитов.

В этом случае мы хотим вернуться в ветке style в точку перед слиянием с main. Нам необходимо найти последний коммит перед слиянием.

Выполните

```
git switch style
git log --graph
```

Результат

```
$ git switch style
Already on 'style'
$ git log --graph
*   79ac6fa 2023-11-28 | Resolved merge conflict (HEAD -> style) [Alexander Shvets]
|\
| * 85c14e9 2023-11-28 | Added meta title (main) [Alexander Shvets]
* | a33deed 2023-11-28 | Merge branch 'main' into style [Alexander Shvets]
|\|
| * ee16740 2023-11-28 | Added README [Alexander Shvets]
* | 0ee0113 2023-11-28 | Renamed hello.html; moved style.css [Alexander Shvets]
* | 903eb1d 2023-11-28 | Included stylesheet into hello.html [Alexander Shvets]
* | 555372e 2023-11-28 | Added css stylesheet [Alexander Shvets]
|/
* 9288a33 2023-11-28 | Added copyright statement with email [Alexander Shvets]
* b7614c1 2023-11-28 | Added HTML header (tag: v1) [Alexander Shvets]
* 46afaff 2023-11-28 | Added standard HTML page tags (tag: v1-beta) [Alexander Shvets]
```

Это немного трудно читать, но, глядя на данные, мы видим, что коммит «Renamed hello.html; moved style.css» был последним на ветке `style` перед слиянием. Давайте сбросим ветку `style` к этому коммиту. Чтобы сослаться на этот коммит, мы либо используем его хеш, либо посчитаем, что этот коммит находится за 2 коммита до `HEAD`, то есть `HEAD~2` в нотации Git.

Выполните

```
git reset --hard HEAD~2
```

Результат

```
$ git reset --hard HEAD~2
HEAD is now at 0ee0113 Renamed hello.html; moved style.css
```

Проверьте ветку

Теперь проверим историю ветки style. В истории не должно быть коммитов слияния.

Выполните

```
git log --graph
```

Результат

```
$ git log --graph
* 0ee0113 2023-11-28 | Renamed hello.html; moved style.css (HEAD -> style) [Alexander Shvets]
* 903eb1d 2023-11-28 | Included stylesheet into hello.html [Alexander Shvets]
* 555372e 2023-11-28 | Added css stylesheet [Alexander Shvets]
* 9288a33 2023-11-28 | Added copyright statement with email [Alexander Shvets]
* b7614c1 2023-11-28 | Added HTML header (tag: v1) [Alexander Shvets]
* 46afaff 2023-11-28 | Added standard HTML page tags (tag: v1-beta) [Alexander Shvets]
* 78433de 2023-11-28 | Added h1 tag [Alexander Shvets]
* 5836970 2023-11-28 | Initial commit [Alexander Shvets]
```

Перебазирование

Мы вернули ветку style к точке перед первым слиянием. При этом в ветке main есть два коммита, которых нет в ветке style: новый файл README и конфликтующее изменение в файле index.html. На этот раз мы перенесем эти изменения в ветку style с помощью команды rebase, а не merge.

Перебазируем ветку style **на ветку** main

Выполните

```
git switch style
git rebase main
git status
```

Результат

```
$ git switch style
Already on 'style'
$ git rebase main
Rebasing (1/3)Rebasing (2/3)Auto-merging hello.html
CONFLICT (content): Merge conflict in hello.html
error: could not apply 903eb1d... Included stylesheet into hello.html
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
Could not apply 903eb1d... Included stylesheet into hello.html
$ git status
interactive rebase in progress; onto 85c14e9
Last commands done (2 commands done):
   pick 555372e Added css stylesheet
   pick 903eb1d Included stylesheet into hello.html
Next command to do (1 remaining command):
   pick 0ee0113 Renamed hello.html; moved style.css
  (use "git rebase --edit-todo" to view and edit)
You are currently rebasing branch 'style' on '85c14e9'.
  (fix conflicts and then run "git rebase --continue")
  (use "git rebase --skip" to skip this patch)
  (use "git rebase --abort" to check out the original branch)

Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
    both modified:   hello.html

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

Опять возник конфликт! Обратите внимание, что конфликт возник в файле hello.html, а не в файле index.html, как в прошлый раз. Это связано с тем, что rebase находился в процессе применения изменений style поверх ветки main. Файл hello.html в main еще не был переименован, поэтому он все еще имеет старое имя. При слиянии возник бы «обратный» конфликт. При слиянии изменения ветки main были бы применены поверх ветки style. В ветке style файл переименован, поэтому конфликт возник бы в файле index.html.

Файл: hello.html

```
<html>
  <head>
<<<<<<< HEAD
    <title>Hello World Page</title>
=======
    <link type="text/css" rel="stylesheet" media="all" href="style.css" />
>>>>>>> 903eb1d (Included stylesheet into hello.html)
  </head>
  <body>
    <h1>Hello, World!</h1>
    <p>Let's learn Git together.</p>
  </body>
</html>
```

Разрешение конфликта

Сам конфликт может быть разрешен тем же способом, что и предыдущий. Сначала мы изменим файл hello.html, чтобы он соответствовал нашим ожиданиям.

Файл: hello.html

```
<html>
  <head>
    <title>Hello World Page</title>
    <link type="text/css" rel="stylesheet" media="all" href="style.css" />
  </head>
  <body>
    <h1>Hello, World!</h1>
    <p>Let's learn Git together.</p>
  </body>
</html>
```

Но после этого нам не нужно коммитить изменения. Мы можем просто добавить файл в индекс и продолжить процесс rebase. Вот почему я люблю rebase! Он позволяет мне устранять конфликты, не создавая кучу уродливых конфликтов слияния.

Для простоты мы можем добавить все изменения, используя ., что означает путь к текущей директории. Git интерпретирует это как «добавить все изменения из текущей директории и её поддиректорий».

Выполните

```
git add .
git rebase --continue
```

Здесь, скорее всего, Git снова откроет редактор, чтобы позволить нам изменить текст коммита. Мы можем оставить текст без изменений. После сохранения изменений Git завершит процесс `rebase`, и мы сможем выполнить следующие команды:

Выполните

```
git status
git log --all --graph
```

Результат

```
$ git add .
$ git rebase --continue
[detached HEAD 23149b5] Included stylesheet into hello.html
 1 file changed, 1 insertion(+)
Rebasing (3/3)[KSuccessfully rebased and updated refs/heads/style.
$ git status
On branch style
nothing to commit, working tree clean
$ git log --all --graph
* 39a1e0f 2023-11-28 | Renamed hello.html; moved style.css (HEAD -> style) [Alexander Shvets]
* 23149b5 2023-11-28 | Included stylesheet into hello.html [Alexander Shvets]
* b9e6de1 2023-11-28 | Added css stylesheet [Alexander Shvets]
* 85c14e9 2023-11-28 | Added meta title (main) [Alexander Shvets]
* ee16740 2023-11-28 | Added README [Alexander Shvets]
* 9288a33 2023-11-28 | Added copyright statement with email [Alexander Shvets]
* b7614c1 2023-11-28 | Added HTML header (tag: v1) [Alexander Shvets]
* 46afaff 2023-11-28 | Added standard HTML page tags (tag: v1-beta) [Alexander Shvets]
* 78433de 2023-11-28 | Added h1 tag [Alexander Shvets]
```

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 main. Однако, дерево коммитов значительно отличается. Дерево коммитов ветки style было переписано таким образом, что ветка main является частью истории коммитов. Это делает цепь коммитов линейной и гораздо более читабельной.

Когда использовать команду rebase**, а когда команду** merge**?**

Используйте команду rebase:

- Когда вы получаете изменения из удаленного репозитория и хотите применить их к своей локальной ветке.
- Если вы хотите, чтобы история коммитов была линейной и легко читаемой.

Не используйте команду rebase:

- Если текущая ветка является публичной и общей. Переписывание таких веток будет мешать работе других членов команды.
- Если важна *точная* история ветки коммитов (поскольку команда rebase переписывает историю коммитов).

Учитывая приведенные выше рекомендации, я предпочитаю использовать команду rebase для краткосрочных, локальных веток и команду merge для веток в публичном репозитории.

Слияние в ветку main

Слейте style **в** main

Выполните

```
git switch main
git merge style
```

Результат

```
$ git switch main
Switched to branch 'main'
$ git merge style
Updating 85c14e9..39a1e0f
Fast-forward
 css/style.css           | 3 +++
 hello.html => index.html | 1 +
 2 files changed, 4 insertions(+)
 create mode 100644 css/style.css
 rename hello.html => index.html (73%)
```

Поскольку последний коммит в main предшествует последнему коммиту ветки style, Git может выполнить ускоренное слияние, просто переместив указатель ветки вперед, на тот же коммит, что и ветка style.

При ускоренном слиянии конфликты не возникают. Кроме того, при ускоренном слиянии не создается фиксация слияния.

Просмотрите логи

Выполните

```
git log --all --graph
```

Результат

```
$ git log --all --graph
* 39a1e0f 2023-11-28 | Renamed hello.html; moved style.css (HEAD -> main, style) [Alexander Shvets]
* 23149b5 2023-11-28 | Included stylesheet into hello.html [Alexander Shvets]
* b9e6de1 2023-11-28 | Added css stylesheet [Alexander Shvets]
* 85c14e9 2023-11-28 | Added meta title [Alexander Shvets]
* ee16740 2023-11-28 | Added README [Alexander Shvets]
* 9288a33 2023-11-28 | Added copyright statement with email [Alexander Shvets]
* b7614c1 2023-11-28 | Added HTML header (tag: v1) [Alexander Shvets]
* 46afaff 2023-11-28 | Added standard HTML page tags (tag: v1-beta) [Alexander Shvets]
* 78433de 2023-11-28 | Added h1 tag [Alexander Shvets]
* 5836970 2023-11-28 | Initial commit [Alexander Shvets]
```

Теперь ветки style и main идентичны.