**Лабораторная работа 4**

**Слияние**

**Цели**: научиться сливать две отличающиеся ветки для переноса изменений обратно в одну ветку.

**1 Слияние веток**

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

**ВЫПОЛНИТЕ:**

git checkout style

git merge master

git hist --all

**РЕЗУЛЬТАТ:**

$ git checkout style

Switched to branch 'style'

$ git merge master

Merge made by recursive.

README | 1 +

1 files changed, 1 insertions(+), 0 deletions(-)

create mode 100644 README

$ git hist --all

\* 5813a3f 2011-03-09 | Merge branch 'master' into style (HEAD, style) [Alexander Shvets]

|\

| \* 6c0f848 2011-03-09 | Added README (master) [Alexander Shvets]

\* | 07a2a46 2011-03-09 | Updated index.html [Alexander Shvets]

\* | 649d26c 2011-03-09 | Hello uses style.css [Alexander Shvets]

\* | 1f3cbd2 2011-03-09 | Added css stylesheet [Alexander Shvets]

|/

\* 8029c07 2011-03-09 | Added index.html. [Alexander Shvets]

\* 567948a 2011-03-09 | Moved hello.html to lib [Alexander Shvets]

\* 6a78635 2011-03-09 | Add an author/email comment [Alexander Shvets]

\* fa3c141 2011-03-09 | Added HTML header (v1) [Alexander Shvets]

\* 8c32287 2011-03-09 | Added standard HTML page tags (v1-beta) [Alexander Shvets]

\* 43628f7 2011-03-09 | Added h1 tag [Alexander Shvets]

\* 911e8c9 2011-03-09 | First Commit [Alexander Shvets]

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.

Однако, это делает графики коммитов действительно уродливыми. Позже мы рассмотрим возможность перебазирования, как альтернативы слиянию.

**2 Далее**

Но что если изменения в ветке master конфликтуют с изменениями в style?

# Создание конфликта

### Цели: создание конфликтующих изменений в ветке master.

## 1 Вернитесь в master и создайте конфликт

Вернитесь в ветку master и внесите следующие изменения:

git checkout master

#### ФАЙЛ: lib/hello.html

<!-- Author: Alexander Shvets (alex@githowto.com) -->

<html>

<head>

**<!-- no style -->**

</head>

<body>

<h1>Hello, World! **Life is great!**</h1>

</body>

</html>

#### ВЫПОЛНИТЕ:

git add lib/hello.html

git commit -m 'Life is great!'

**Внимание:** используйте для этого коммита одинарные кавычки, дабы избежать проблем с символом !. В bash он считается служебным.

## 2 Просмотр веток

#### ВЫПОЛНИТЕ:

git hist --all

#### РЕЗУЛЬТАТ:

$ git hist --all

\* 454ec68 2011-03-09 | Life is great! (HEAD, master) [Alexander Shvets]

| \* 5813a3f 2011-03-09 | Merge branch 'master' into style (style) [Alexander Shvets]

| |\

| |/

|/|

\* | 6c0f848 2011-03-09 | Added README [Alexander Shvets]

| \* 07a2a46 2011-03-09 | Updated index.html [Alexander Shvets]

| \* 649d26c 2011-03-09 | Hello uses style.css [Alexander Shvets]

| \* 1f3cbd2 2011-03-09 | Added css stylesheet [Alexander Shvets]

|/

\* 8029c07 2011-03-09 | Added index.html. [Alexander Shvets]

\* 567948a 2011-03-09 | Moved hello.html to lib [Alexander Shvets]

\* 6a78635 2011-03-09 | Add an author/email comment [Alexander Shvets]

\* fa3c141 2011-03-09 | Added HTML header (v1) [Alexander Shvets]

\* 8c32287 2011-03-09 | Added standard HTML page tags (v1-beta) [Alexander Shvets]

\* 43628f7 2011-03-09 | Added h1 tag [Alexander Shvets]

\* 911e8c9 2011-03-09 | First Commit [Alexander Shvets]

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.

## 3 Далее

Последнее изменение в master конфликтует с некоторыми изменениями в style. На следующем шаге мы решим этот конфликт.

# Разрешение конфликтов

### Цели: научиться разрешать конфликты во время слияния

## 1 Слияние master с веткой style

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

#### ВЫПОЛНИТЕ:

git checkout style

git merge master

#### РЕЗУЛЬТАТ:

$ git checkout style

Switched to branch '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.

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

#### ФАЙЛ: lib/hello.html

<!-- Author: Alexander Shvets (alex@githowto.com) -->

<html>

<head>

<<<<<<< HEAD

<link type="text/css" rel="stylesheet" media="all" href="style.css" />

=======

<!-- no style -->

>>>>>>> master

</head>

<body>

<h1>Hello,World! Life is great!</h1>

</body>

</html>

Первый раздел - версия во главе текущей ветки (style). Второй раздел - версия ветки master.

## 2 Решение конфликта

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

#### ФАЙЛ: lib/hello.html

<!-- Author: Alexander Shvets (alex@githowto.com) -->

<html>

<head>

<link type="text/css" rel="stylesheet" media="all" href="style.css" />

</head>

<body>

<h1>Hello, World! Life is great!</h1>

</body>

</html>

## 3 Сделайте коммит решения конфликта

#### ВЫПОЛНИТЕ:

git add lib/hello.html

git commit -m "Merged master fixed conflict."

#### РЕЗУЛЬТАТ:

$ git add lib/hello.html

$ git commit -m "Merged master fixed conflict."

Recorded resolution for 'lib/hello.html'.

[style 645c4e6] Merged master fixed conflict.

## 4 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](https://stackoverflow.com/questions/137102/whats-the-best-visual-merge-tool-for-git)).

# Перебазирование как альтернатива слиянию

### Цели: Узнать различия между перебазированием и слиянием.

### 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

### Цели: Сброс ветки style до точки перед первым слиянием.

## 1 Сброс ветки style

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

В этом случае мы хотим вернуться в стиль в точку перед слиянием с мастером. Нам необходимо найти последний коммит перед слиянием.

#### ВЫПОЛНИТЕ:

стиль оформления заказа git

git hist

#### РЕЗУЛЬТАТ:

$ git стиль оформления заказа

Уже в стиле

$ git hist

\* 645c4e6 09.03.2011 | Исправлен конфликт объединенного мастера. (ГОЛОВА, стиль) [Александр Швец]

| \

| \* 454ec68 09.03.2011 | Жизнь замечательна! (мастер) [Александр Швец]

\* | 5813a3f 09.03.2011 | Объединить ветку master в стиль [Александр Швец]

| \ \

| | /

| \* 6c0f848 09.03.2011 | Добавлен README [Александр Швец]

\* | 07a2a46 09.03.2011 | Обновлен index.html [Александр Швец]

\* | 649d26c 09.03.2011 | Hello использует style.css [Александр Швец]

\* | 1f3cbd2 09.03.2011 | Добавлена ​​таблица стилей css [Александр Швец]

| /

\* 8029c07 09.03.2011 | Добавлен index.html. [Александр Швец]

\* 567948a 09.03.2011 | Файл hello.html перенесен в библиотеку [Александр Швец]

\* 6a78635 09.03.2011 | Добавить комментарий автора / адрес электронной почты [Александр Швец]

\* fa3c141 09.03.2011 | Добавлен HTML-заголовок (v1) [Александр Швец]

\* 8c32287 09.03.2011 | Добавлены стандартные теги HTML-страницы (v1-beta) [Александр Швец]

\* 43628f7 09.03.2011 | Добавлен тег h1 [Александр Швец]

\* 911e8c9 09.03.2011 | Первый коммит [Александр Швец]

Это немного трудно читать, но, глядя на данные, мы видим, что коммит «Обновлен index.html» был последним на ветке стиль перед слиянием. Давайте сбросим ветку стиль к этому коммиту.

#### ВЫПОЛНИТЕ:

git reset --hard <хэш>

#### РЕЗУЛЬТАТ:

$ git reset --hard 07a2a46

HEAD теперь 07a2a46 Обновлен index.html

## 

## 2 Проверьте ветку.

Поищите логки стиля. У нас в истории больше нет коммитов слияний.

#### ВЫПОЛНИТЕ:

git hist --all

#### РЕЗУЛЬТАТ:

$ git hist --all

\* 454ec68 09.03.2011 | Жизнь замечательна! (мастер) [Александр Швец]

\* 6c0f848 09.03.2011 | Добавлен README [Александр Швец]

| \* 07a2a46 09.03.2011 | Обновлен index.html (HEAD, style) [Александр Швец]

| \* 649d26c 09.03.2011 | Hello использует style.css [Александр Швец]

| \* 1f3cbd2 09.03.2011 | Добавлена ​​таблица стилей css [Александр Швец]

| /

\* 8029c07 09.03.2011 | Добавлен index.html. [Александр Швец]

\* 567948a 09.03.2011 | Файл hello.html перенесен в библиотеку [Александр Швец]

\* 6a78635 09.03.2011 | Добавить комментарий автора / адрес электронной почты [Александр Швец]

\* fa3c141 09.03.2011 | Добавлен HTML-заголовок (v1) [Александр Швец]

\* 8c32287 09.03.2011 | Добавлены стандартные теги HTML-страницы (v1-beta) [Александр Швец]

\* 43628f7 09.03.2011 | Добавлен тег h1 [Александр Швец]

\* 911e8c9 09.03.2011 | Первый коммит [Александр Швец]

# Сброс ветки master

### Цели: сбросить ветку master в точку до конфликтующего коммита.

## 1 Сброс ветки master

Добавив интерактивный режим в ветку master, мы внесли изменения, конфликтующие с изменениями в ветке style. Давайте вернемся в ветке master в точку перед внесением конфликтующих изменений. Это позволяет нам продемонстрировать работу команды rebase, не беспокоясь о конфликтах.

#### ВЫПОЛНИТЕ:

git checkout master

git hist

#### РЕЗУЛЬТАТ:

$ git hist

\* 454ec68 2011-03-09 | Life is great! (HEAD, master) [Alexander Shvets]

\* 6c0f848 2011-03-09 | Added README [Alexander Shvets]

\* 8029c07 2011-03-09 | Added index.html. [Alexander Shvets]

\* 567948a 2011-03-09 | Moved hello.html to lib [Alexander Shvets]

\* 6a78635 2011-03-09 | Add an author/email comment [Alexander Shvets]

\* fa3c141 2011-03-09 | Added HTML header (v1) [Alexander Shvets]

\* 8c32287 2011-03-09 | Added standard HTML page tags (v1-beta) [Alexander Shvets]

\* 43628f7 2011-03-09 | Added h1 tag [Alexander Shvets]

\* 911e8c9 2011-03-09 | First Commit [Alexander Shvets]

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

#### ВЫПОЛНИТЕ:

git reset --hard <hash>

git hist --all

Просмотрите лог. Он должен выглядеть, как будто репозиторий был перемотан назад во времени к точке до какого-либо слияния.

#### РЕЗУЛЬТАТ:

$ git reset --hard 6c0f848

$ git hist --all

\* 6c0f848 2011-03-09 | Added README (HEAD, master) [Alexander Shvets]

| \* 07a2a46 2011-03-09 | Updated index.html (style) [Alexander Shvets]

| \* 649d26c 2011-03-09 | Hello uses style.css [Alexander Shvets]

| \* 1f3cbd2 2011-03-09 | Added css stylesheet [Alexander Shvets]

|/

\* 8029c07 2011-03-09 | Added index.html. [Alexander Shvets]

\* 567948a 2011-03-09 | Moved hello.html to lib [Alexander Shvets]

\* 6a78635 2011-03-09 | Add an author/email comment [Alexander Shvets]

\* fa3c141 2011-03-09 | Added HTML header (v1) [Alexander Shvets]

\* 8c32287 2011-03-09 | Added standard HTML page tags (v1-beta) [Alexander Shvets]

\* 43628f7 2011-03-09 | Added h1 tag [Alexander Shvets]

\* 911e8c9 2011-03-09 | First Commit [Alexander Shvets]

# Перебазирование

### Цели: использовать команду rebase вместо команды merge.

Итак, 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 rebase вместо слияния.

#### ВЫПОЛНИТЕ:

git checkout style

git rebase master

git hist

#### РЕЗУЛЬТАТ:

$ go style

Switched to branch 'style'

$

$ git rebase master

First, rewinding head to replay your work on top of it...

Applying: Added css stylesheet

Applying: Hello uses style.css

Applying: Updated index.html

$

$ git hist

\* 6e6c76a 2011-03-09 | Updated index.html (HEAD, style) [Alexander Shvets]

\* 1436f13 2011-03-09 | Hello uses style.css [Alexander Shvets]

\* 59da9a7 2011-03-09 | Added css stylesheet [Alexander Shvets]

\* 6c0f848 2011-03-09 | Added README (master) [Alexander Shvets]

\* 8029c07 2011-03-09 | Added index.html. [Alexander Shvets]

\* 567948a 2011-03-09 | Moved hello.html to lib [Alexander Shvets]

\* 6a78635 2011-03-09 | Add an author/email comment [Alexander Shvets]

\* fa3c141 2011-03-09 | Added HTML header (v1) [Alexander Shvets]

\* 8c32287 2011-03-09 | Added standard HTML page tags (v1-beta) [Alexander Shvets]

\* 43628f7 2011-03-09 | Added h1 tag [Alexander Shvets]

\* 911e8c9 2011-03-09 | First Commit [Alexander Shvets]

## 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.

## 2 Когда использовать перебазирование, а когда слияние?

Не используйте перебазирование …

1. Если ветка является публичной и расшаренной. Переписывание общих веток будет мешать работе других членов команды.
2. Когда важна точная история коммитов ветки (так как команда rebase переписывает историю коммитов).

Учитывая приведенные выше рекомендации, я предпочитаю использовать rebase для кратковременных, локальных веток, а слияние для веток в публичном репозитории.

# Слияние в ветку master

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

## 1 Слияние style в master

#### ВЫПОЛНИТЕ:

git checkout master

git merge style

#### РЕЗУЛЬТАТ:

$ git checkout master

Switched to branch 'master'

$

$ git merge style

Updating 6c0f848..6e6c76a

Fast-forward

index.html | 2 +-

lib/style.css | 8 ++++++++

lib/hello.html | 6 ++++--

3 files changed, 13 insertions(+), 3 deletions(-)

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.

## 2 Просмотрите логи

#### ВЫПОЛНИТЕ:

git hist

#### РЕЗУЛЬТАТ:

$ git hist

\* 6e6c76a 2011-03-09 | Updated index.html (HEAD, master, style) [Alexander Shvets]

\* 1436f13 2011-03-09 | Hello uses style.css [Alexander Shvets]

\* 59da9a7 2011-03-09 | Added css stylesheet [Alexander Shvets]

\* 6c0f848 2011-03-09 | Added README [Alexander Shvets]

\* 8029c07 2011-03-09 | Added index.html. [Alexander Shvets]

\* 567948a 2011-03-09 | Moved hello.html to lib [Alexander Shvets]

\* 6a78635 2011-03-09 | Add an author/email comment [Alexander Shvets]

\* fa3c141 2011-03-09 | Added HTML header (v1) [Alexander Shvets]

\* 8c32287 2011-03-09 | Added standard HTML page tags (v1-beta) [Alexander Shvets]

\* 43628f7 2011-03-09 | Added h1 tag [Alexander Shvets]

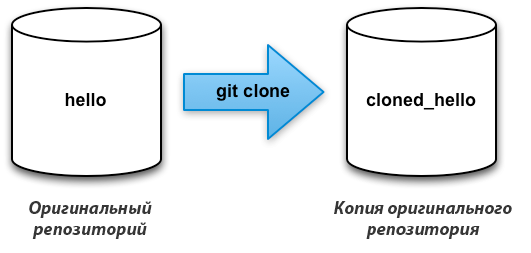
\* 911e8c9 2011-03-09 | First Commit [Alexander Shvets]

Теперь ветки style и master идентичны.

# 36. Несколько репозиториев

До сих пор мы работали с одним git репозиторием. Однако, git удается отлично работать с несколькими репозиториями. Эти дополнительные репозитории могут храниться локально, или доступ к ним может осуществляться через сетевое подключение.

В следующем разделе мы создадим новый репозиторий с именем «cloned\_hello». Мы покажем, как перемещать изменения из одного репозитория в другой и как разрешать конфликты, возникающие в результате работы с двумя репозиториями.



А пока что поработаем с локальными репозиториями (т.е. репозиториями, хранящимися на вашем локальном жестком диске), однако практически все, что вы узнаете в этом разделе, будет применяться к нескольким репозиториям, несмотря на то, хранятся ли они локально или являются публичными.

**Примечание:** Мы будем вносить изменения в обе копии наших репозиториев. Обращайте внимание на то, в каком репозитории вы находитесь на каждом шаге в следующих уроках.

# Клонирование репозиториев

### Цели: Научиться делать копии репозиториев.

Если вы работаете в команде, последующие 12 глав довольно важны в понимании, т.к. вы почти всегда будете работать с клонированными репозиториями.

## 1 Перейдите в рабочий каталог

Перейдите в рабочий каталог и сделайте клон вашего репозитория hello.

#### ВЫПОЛНИТЕ:

cd ..

pwd

ls

**Примечание: Сейчас мы находимся в рабочем каталоге.**

#### РЕЗУЛЬТАТ:

$ cd ..

$ pwd

/Users/alex/Documents/Presentations/githowto/auto

$ ls

hello

В этот момент вы должны находиться в «рабочем» каталоге. Здесь должен быть единственный репозиторий под названием «hello».

## 2 Создайте клон репозитория hello

Давайте создадим клон репозитория.

#### ВЫПОЛНИТЕ:

git clone hello cloned\_hello

ls

#### РЕЗУЛЬТАТ:

$ git clone hello cloned\_hello

Cloning into cloned\_hello...

done.

$ ls

cloned\_hello

hello

В вашем рабочем каталоге теперь должно быть два репозитория: оригинальный репозиторий «hello» и клонированный репозиторий «cloned\_hello»

# Просмотр клонированного репозитория

### Цели: узнать о ветках в удаленных репозиториях.

## 1 Посмотрите на клонированный репозиторий

Давайте взглянем на клонированный репозиторий.

#### ВЫПОЛНИТЕ:

cd cloned\_hello

ls

#### РЕЗУЛЬТАТ:

$ cd cloned\_hello

$ ls

README

index.html

lib

Вы увидите список всех файлов на верхнем уровне оригинального репозитория README, index.html и lib).

## 2 Просмотрите историю репозитория

#### ВЫПОЛНИТЕ:

git hist --all

#### РЕЗУЛЬТАТ:

$ git hist --all

\* 6e6c76a 2011-03-09 | Updated index.html (HEAD, origin/master, origin/style, origin/HEAD, master) [Alexander Shvets]

\* 1436f13 2011-03-09 | Hello uses style.css [Alexander Shvets]

\* 59da9a7 2011-03-09 | Added css stylesheet [Alexander Shvets]

\* 6c0f848 2011-03-09 | Added README [Alexander Shvets]

\* 8029c07 2011-03-09 | Added index.html. [Alexander Shvets]

\* 567948a 2011-03-09 | Moved hello.html to lib [Alexander Shvets]

\* 6a78635 2011-03-09 | Add an author/email comment [Alexander Shvets]

\* fa3c141 2011-03-09 | Added HTML header (v1) [Alexander Shvets]

\* 8c32287 2011-03-09 | Added standard HTML page tags (v1-beta) [Alexander Shvets]

\* 43628f7 2011-03-09 | Added h1 tag [Alexander Shvets]

\* 911e8c9 2011-03-09 | First Commit [Alexander Shvets]

Вы увидите список всех коммитов в новый репозиторий, и он должен (более или менее) совпадать с историей коммитов в оригинальном репозитории. Единственная разница должна быть в названиях веток.

## 3 Удаленные ветки

Вы увидите ветку **master** (**HEAD**) в списке истории. Вы также увидите ветки со странными именами (**origin/master**, **origin/style** и **origin/HEAD**). Мы поговорим о них чуть позже.

# Что такое origin?

### Цели: Узнать об именах удаленных репозиториев.

#### ВЫПОЛНИТЕ:

git remote

#### РЕЗУЛЬТАТ:

$ git remote

origin

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:

#### ВЫПОЛНИТЕ:

git remote show origin

#### РЕЗУЛЬТАТ:

$ git remote show origin

\* remote origin

Fetch URL: /Users/alex/Documents/Presentations/githowto/auto/hello

Push URL: /Users/alex/Documents/Presentations/githowto/auto/hello

HEAD branch (remote HEAD is ambiguous, may be one of the following):

style

master

Remote branches:

style tracked

master 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)

Мы видим, что «имя по умолчанию»(«origin») удаленного репозитория – оригинальное **hello**. 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).

# Удаленные ветки

### Цели:узнать о локальных и удаленных ветках

Давайте посмотрим на ветки, доступные в нашем клонированном репозитории.

#### ВЫПОЛНИТЕ:

git branch

#### РЕЗУЛЬТАТ:

$ git branch

\* master

Как мы видим, в списке только ветка master. Где ветка style? Команда **git** **branch** выводит только список локальных веток по умолчанию.

## 1 Список удаленных веток

Для того, чтобы увидеть все ветки, попробуйте следующую команду:

#### ВЫПОЛНИТЕ:

git branch -a

#### РЕЗУЛЬТАТ:

$ git branch -a

\* master

remotes/origin/HEAD -> origin/master

remotes/origin/style

remotes/origin/master

Git выводит все коммиты в оригинальный репозиторий, но ветки в удаленном репозитории не рассматриваются как локальные. Если мы хотим иметь собственную ветку **style**, мы должны сами ее создать. Через минуту вы увидите, как это делается.