

Технологии разработки программного обеспечения

2020 / 2021, 1 курс, 2 семестр

Пудов Сергей Григорьевич

Лекция 5

- ▶ Git на локальной машине:
 - ▶ refresh
 - ▶ rebase
- ▶ Работа с удалёнными репозиториями
- ▶ Удаленные ветки
- ▶ Отслеживание веток

Git на локальной машине: повтор

Книга: <https://git-scm.com/book/ru/v2>

Помощь:

- ▶ `$ git help <verb>`
- ▶ `$ git <verb> --help`
- ▶ `$ man git-<verb>`

Создание репозитория:

- ▶ `$ git init`

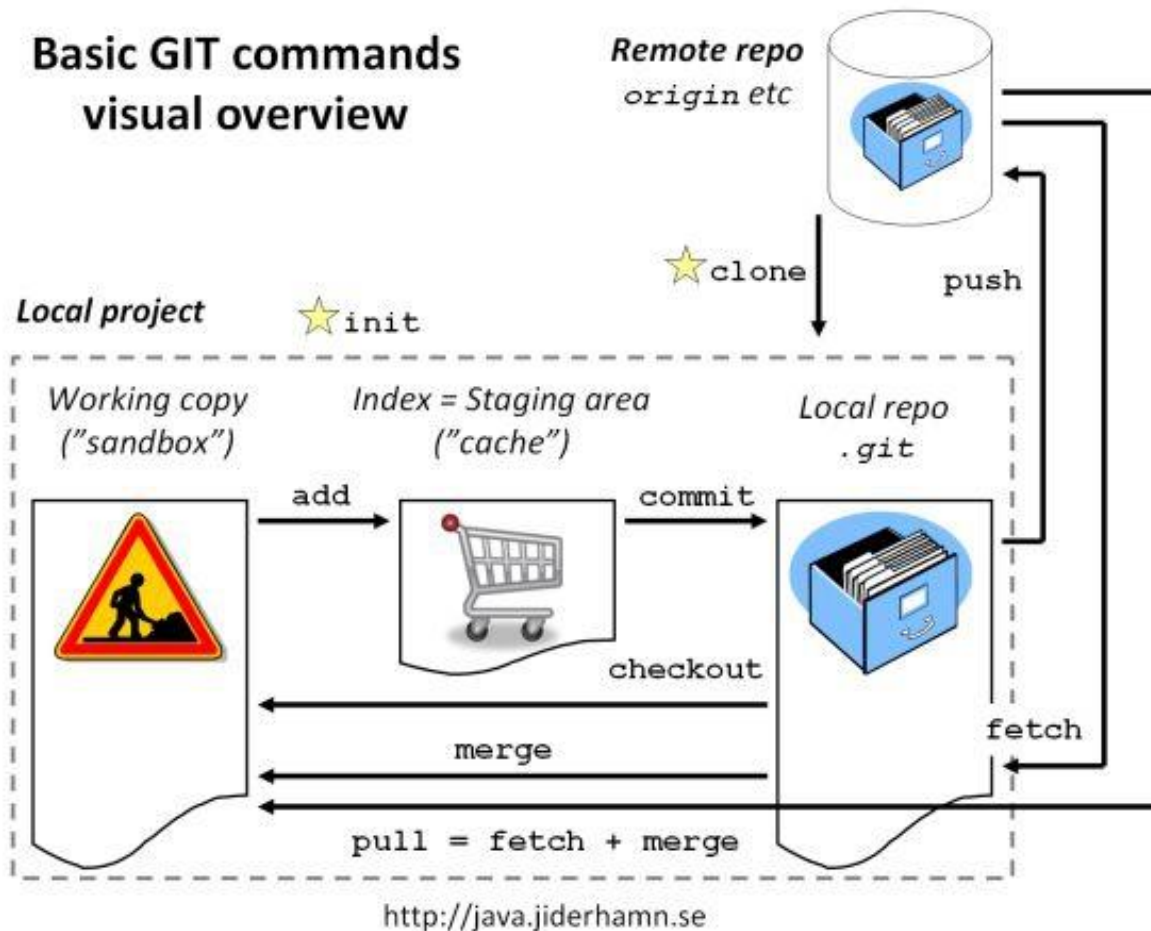
Клонирование удаленного репозитория

- ▶ `$ git clone`

Добавление файлов под версионный контроль:

- ▶ `$ git add <filename>`
- ▶ `$ git commit -m 'initial project version'`
- ▶ `$ git push [remote-name] [branch-name]`
- ▶ `$ git push origin master`

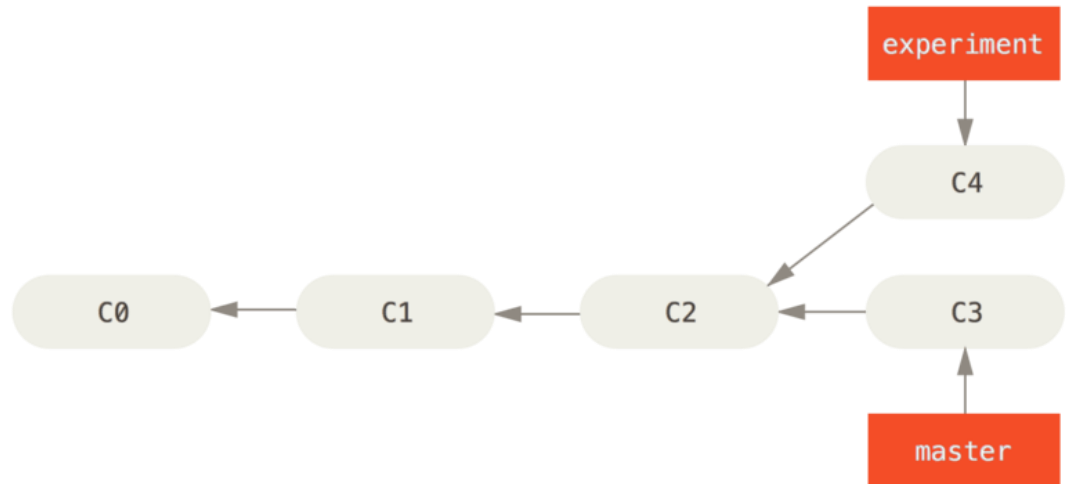
Git на локальной машине: повтор



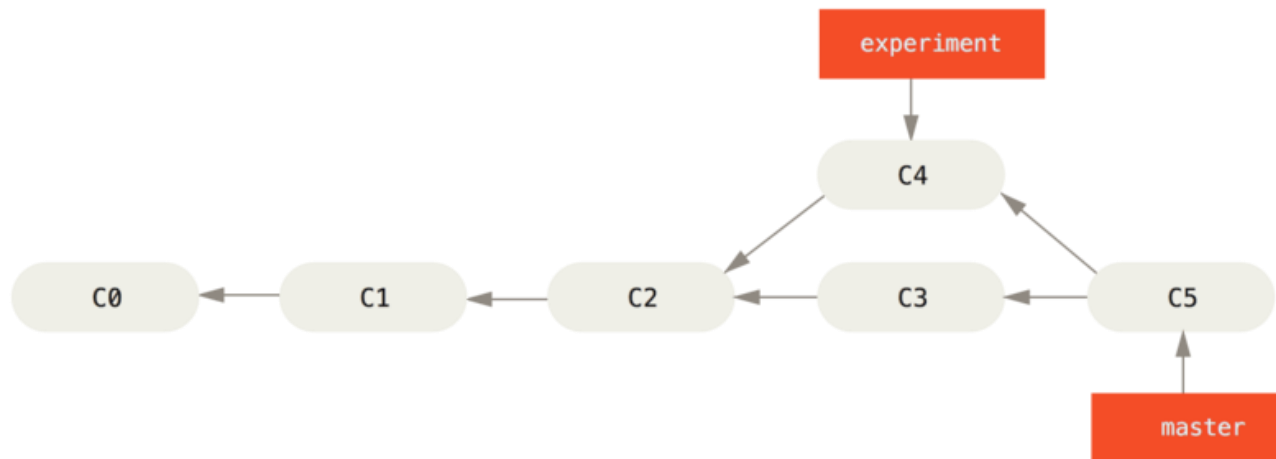
<http://www.about-dev.com/version-control/44-git-intro>

Git на локальной машине: rebase

История до слияния:

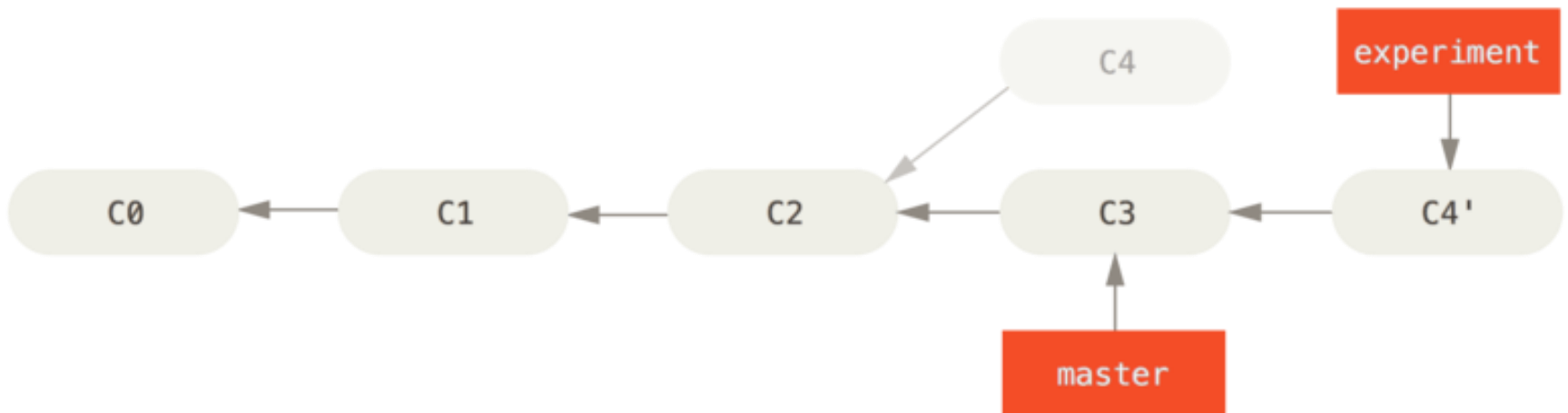


После слияния:



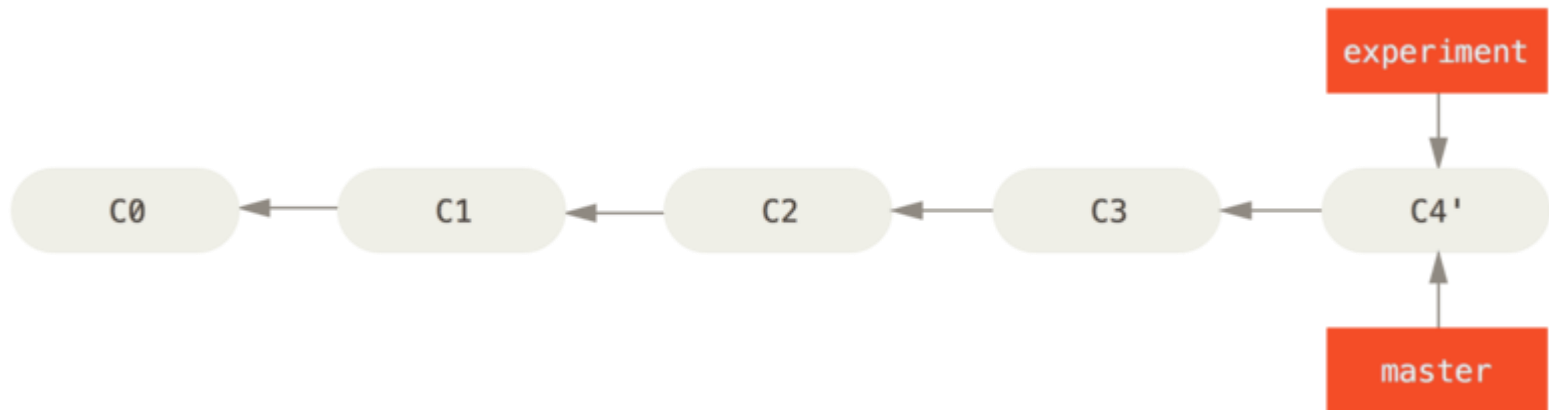
Git на локальной машине: rebase

- ▶ Можно взять те изменения, что были представлены в C4 и применить их поверх C3. В Git это называется *перебазированием* (rebasing).
- ▶ **\$ git checkout experiment**
- ▶ **\$ git rebase master**

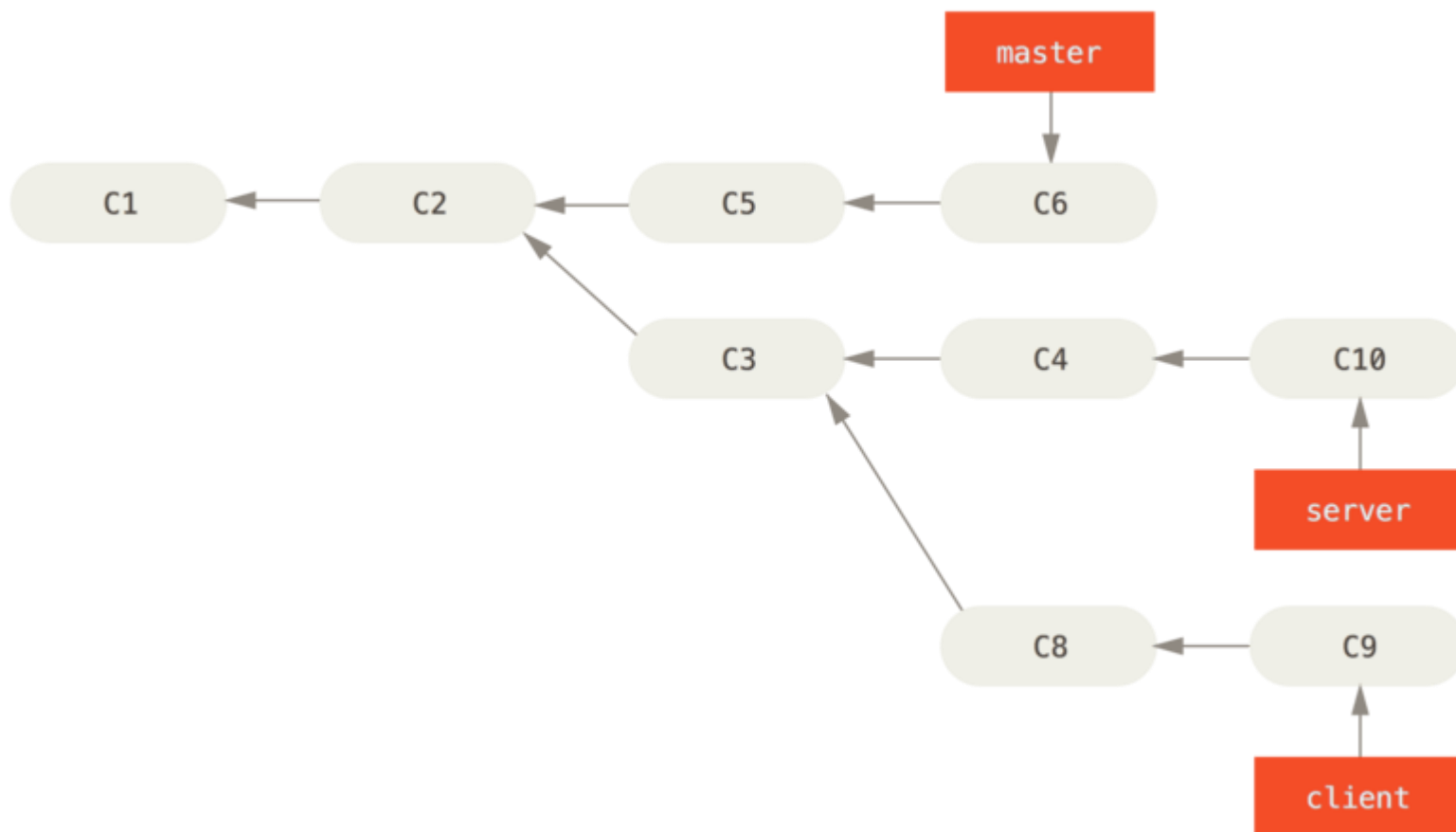


Git на локальной машине: rebase

- ▶ На этом моменте можно переключиться обратно на ветку master и выполнить слияние перемоткой.
- ▶ ***\$ git checkout master***
- ▶ ***\$ git merge experiment***

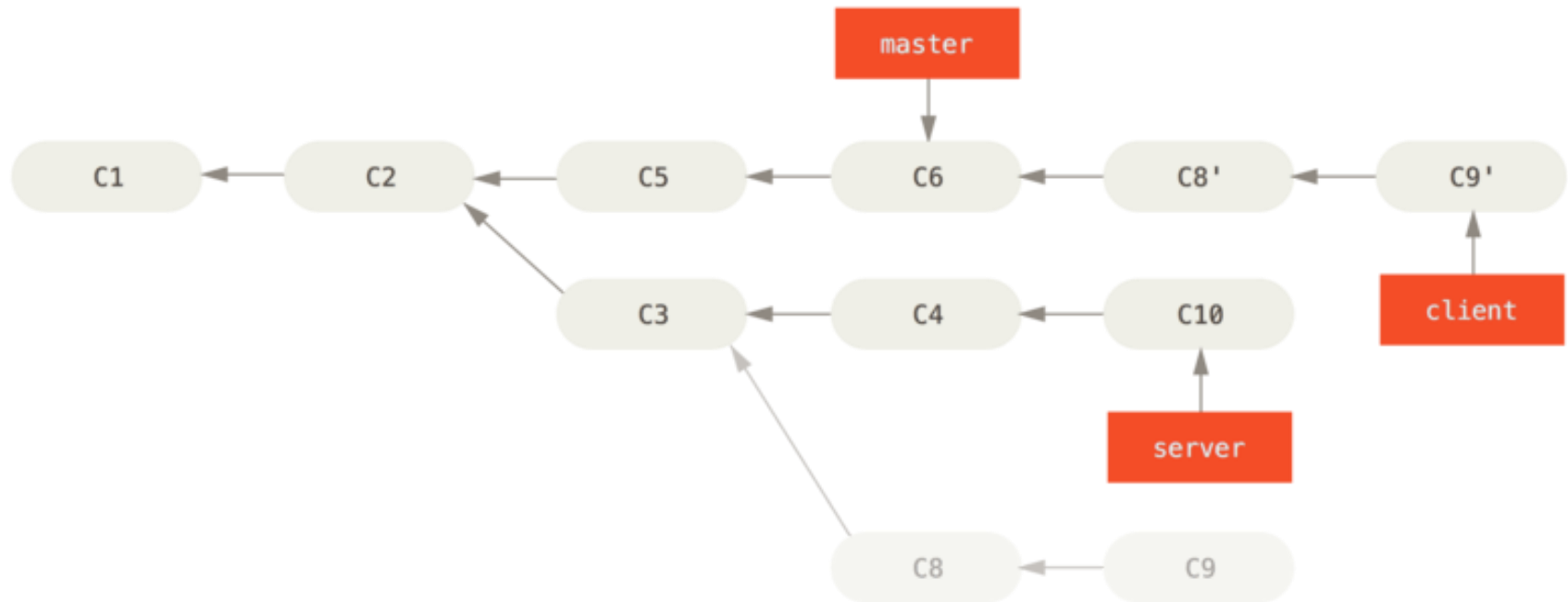


Git на локальной машине: rebase



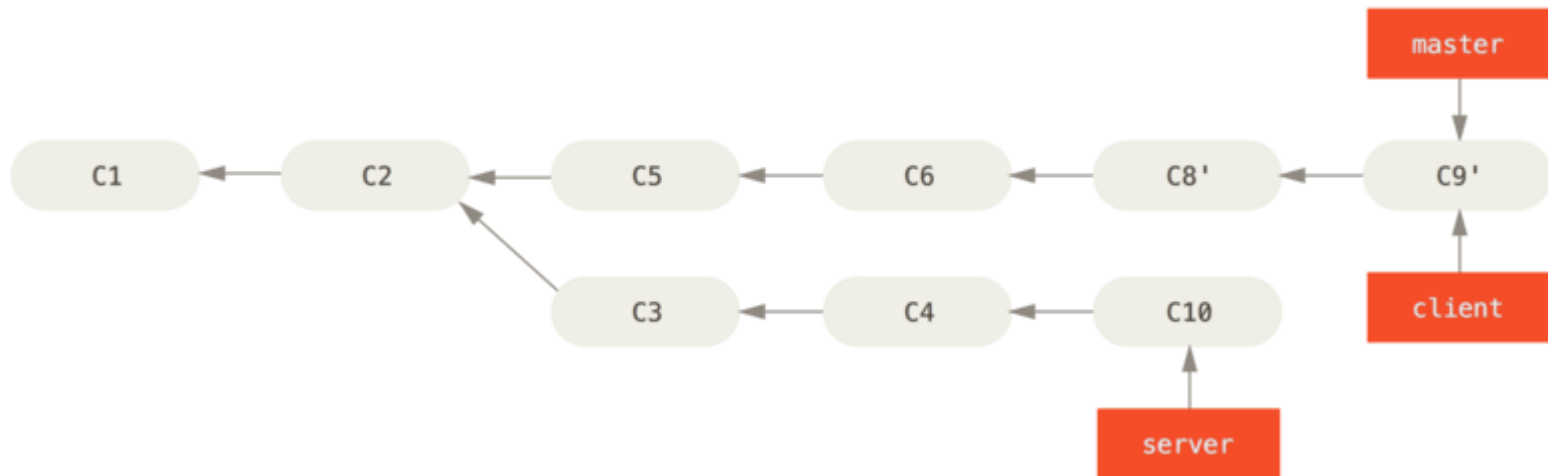
Git на локальной машине: rebase

- ▶ **`$ git rebase --onto master server client`**



Git на локальной машине: rebase

- ▶ **\$ git checkout master**
- ▶ **\$ git merge client**



Работа с удалёнными репозиториями

Чтобы просмотреть список настроенных удалённых репозиториях, вы можете запустить команду `git remote`. Она выведет названия доступных удалённых репозиториях. Если вы клонировали репозиторий, то увидите как минимум `origin` — имя по умолчанию для исходного репозитория:

▶ **`$ git clone https://github.com/schacon/ticgit`**

Cloning into 'ticgit'...

remote: Reusing existing pack: 1857, done.

remote: Total 1857 (delta 0), reused 0 (delta 0)

Receiving objects: 100% (1857/1857), 374.35 KiB | 268.00 KiB/s, done.

Resolving deltas: 100% (772/772), done.

Checking connectivity... done.

▶ **`$ cd ticgit`**

▶ **`$ git remote`**

origin

Работа с удалёнными репозиториями

▶ **\$ git remote -v**

origin <https://github.com/schacon/ticgit> (fetch)

origin <https://github.com/schacon/ticgit> (push)

Для того, чтобы добавить удалённый репозиторий и присвоить ему имя (shortname), просто выполните команду ***git remote add [shortname] [url]***:

▶ **\$ git remote add pb <https://github.com/paulboone/ticgit>**

▶ **\$ git remote -v**

origin <https://github.com/schacon/ticgit> (fetch)

origin <https://github.com/schacon/ticgit> (push)

pb <https://github.com/paulboone/ticgit> (fetch)

pb <https://github.com/paulboone/ticgit> (push)

Работа с удалёнными репозиториями

Для получения данных из удалённых проектов, следует выполнить:

- ▶ ***\$ git fetch [remote-name]***

Или

- ▶ ***\$ git pull [remote-name] = git fetch + git merge***

Когда вы хотите поделиться своими наработками, вам необходимо отправить (push) их в главный репозиторий. Команда для этого действия простая:

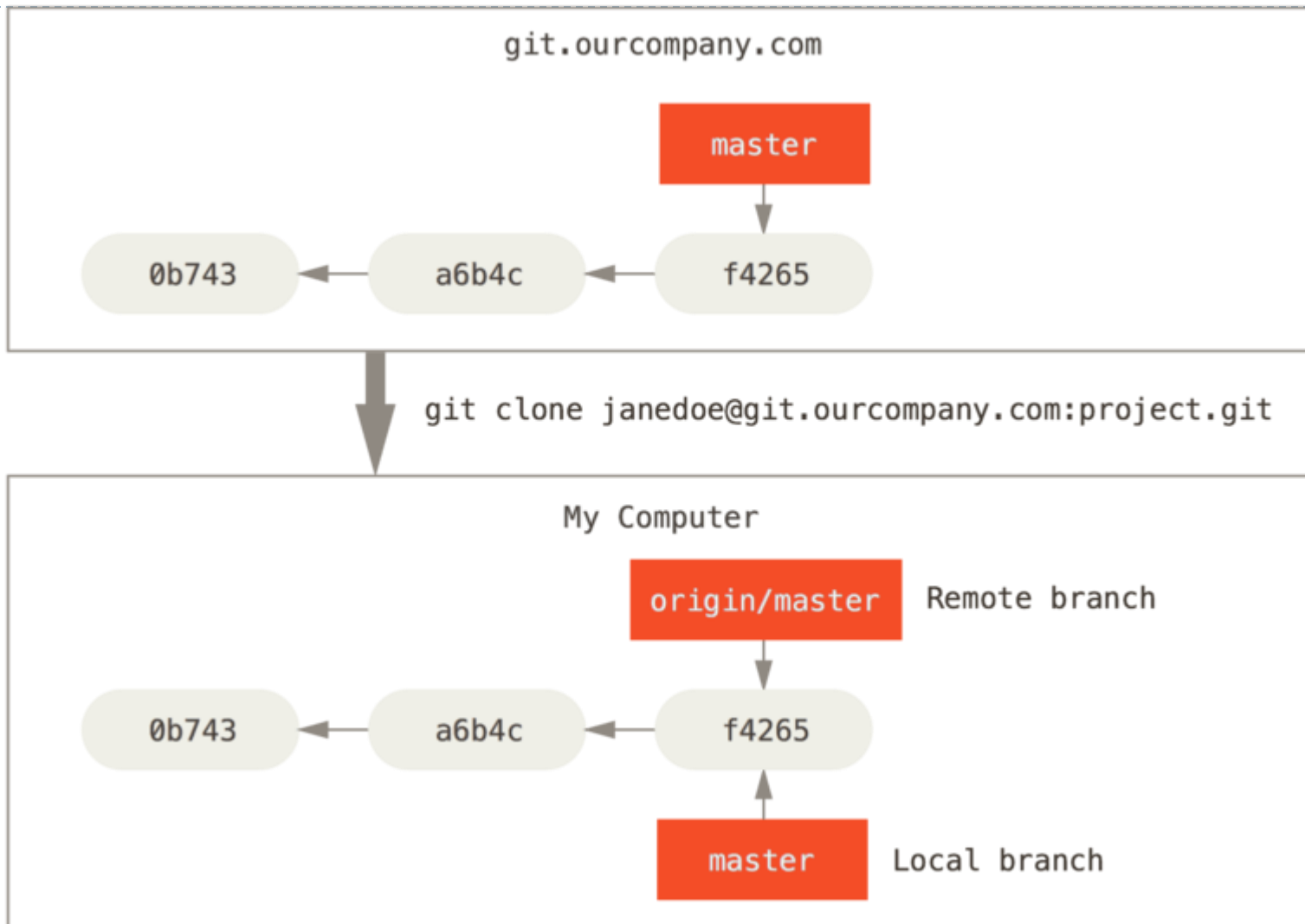
- ▶ ***\$ git push [remote-name] [branch-name]***

Удаленные ветки

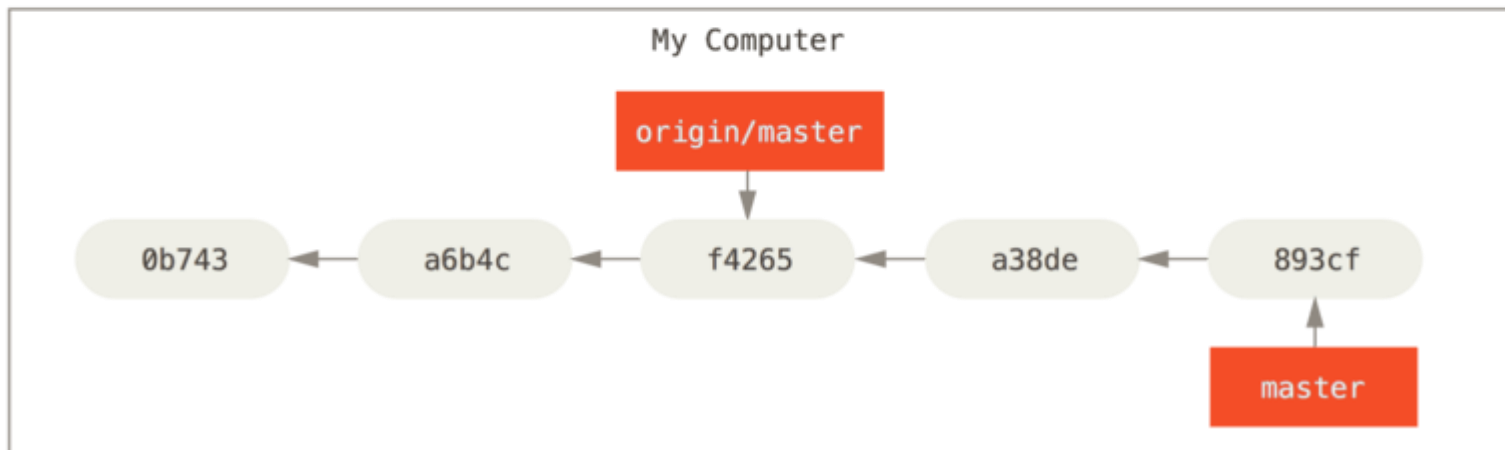
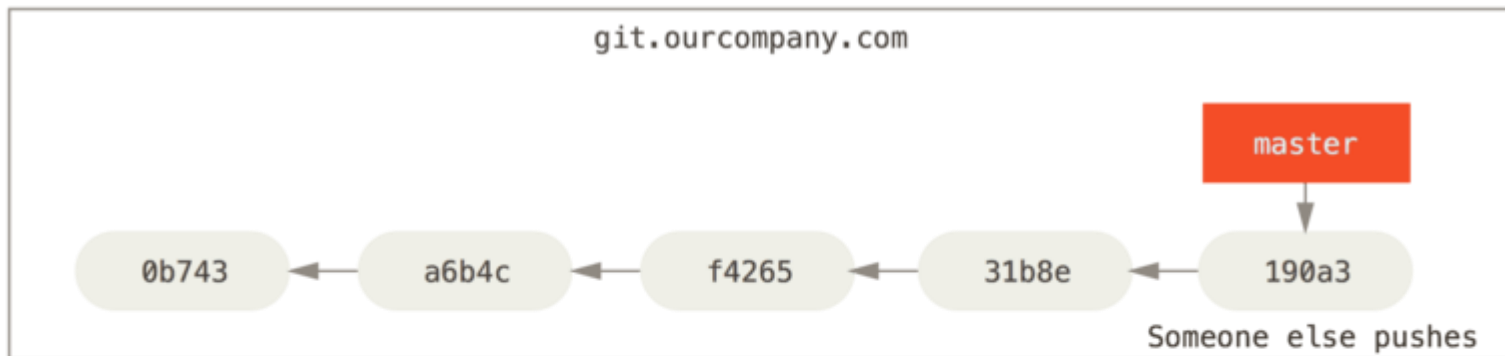
Удалённые ветки — это ссылки (pointers) на состояние веток в ваших удалённых репозиториях. Это локальные ветки, которые нельзя перемещать; они двигаются автоматически всякий раз, когда вы осуществляете связь по сети. Удалённые ветки действуют как закладки для напоминания о том, где ветки в удалённых репозиториях находились во время последнего подключения к ним.

Они выглядят как *(имя удал. репоз.)/(ветка)*. Например, если вы хотите посмотреть, как выглядела ветка **master** на сервере **origin** во время последнего соединения с ним, проверьте ветку **origin/master**.

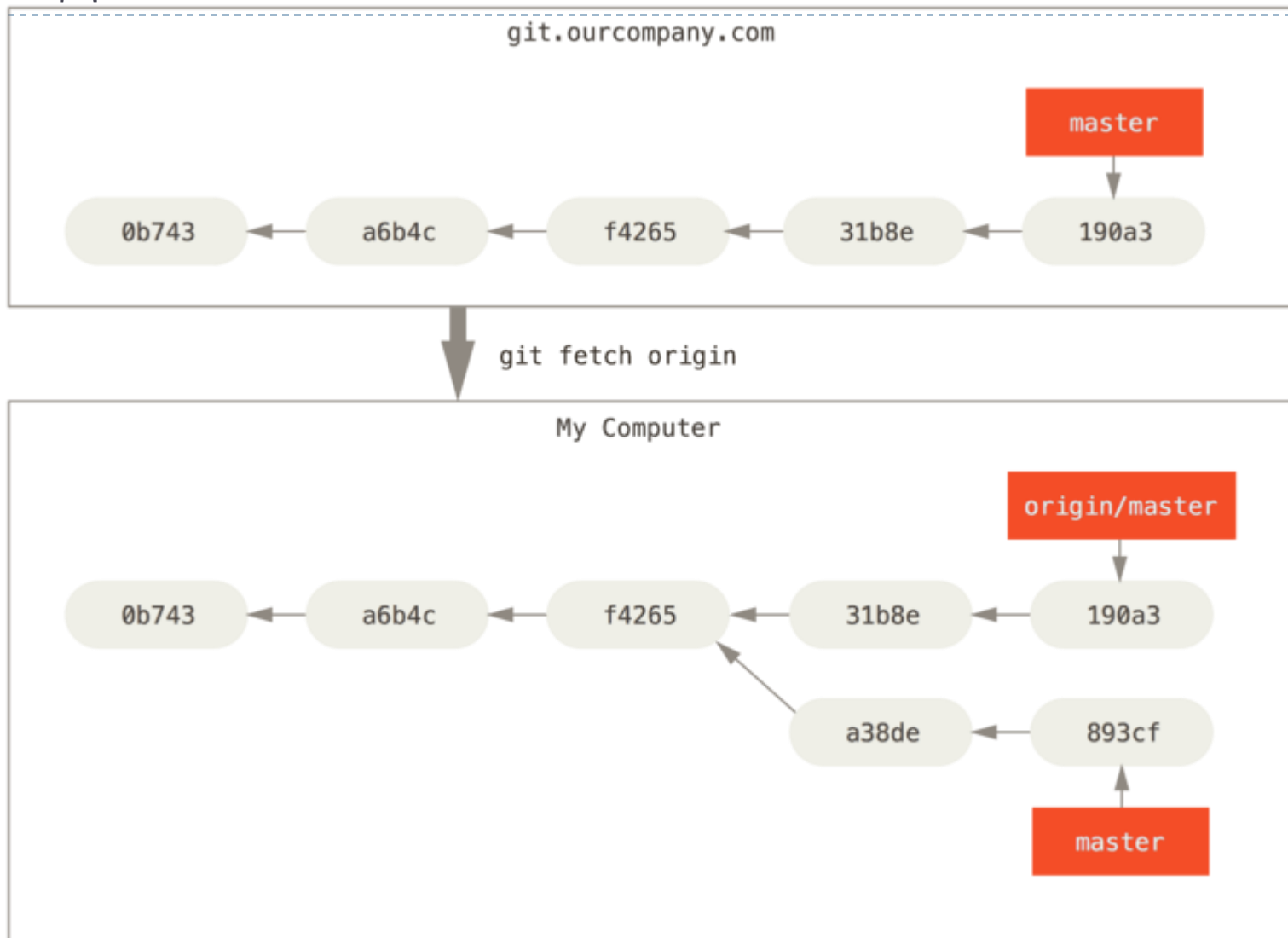
Удаленные ветки



Удаленные ветки



Удаленные ветки



Git: push / fetch

- ▶ Локальные ветки автоматически не синхронизируются с удалёнными серверами — нужно явно отправить те ветки, которыми хотите поделиться.
- ▶ Если есть ветка `serverfix`, над которой хотите работать с кем-то ещё, то можете отправить её, выполнив `git push (удал. сервер) (ветка)`:
- ▶ ***\$ git push origin serverfix***
- ▶ Когда один из ваших соавторов будет получать обновления с сервера, он получит ссылку на то, на что указывает `serverfix` на сервере, как удалённую ветку `origin/serverfix`:
- ▶ ***\$ git fetch origin***

From <https://github.com/schacon/simplegit>

* [new branch] `serverfix` -> `origin/serverfix`

Но: не получите новую ветку `serverfix` — только указатель `origin/serverfix`, который не сможете менять.

Отслеживание веток

- ▶ Отслеживаемые ветки — это локальные ветки, которые напрямую связаны с удалённой веткой.

- ▶ **`$git checkout -b [ветка] [удал. сервер]/[ветка]`**

- ▶ **`$ git checkout --track origin/serverfix`**

Branch serverfix set up to track remote branch serverfix from origin.

Switched to a new branch 'serverfix'

- ▶ **`$ git checkout -b sf origin/serverfix`**

Branch sf set up to track remote branch serverfix from origin.

Switched to a new branch 'sf'

Отслеживание веток

- ▶ `$ git branch -vv`
 - ▶ `iss53 7e424c3 [origin/iss53: ahead 2] forgot the brackets`
 - ▶ `master 1ae2a45 [origin/master] deploying index fix`
 - ▶ `* serverfix f8674d9 [teamone/server-fix-good: ahead 3, behind 1] this should do it`
 - ▶ `testing 5ea463a trying something new`
-
- ▶ `iss53`-ветка отслеживает `origin/iss53` и она опережает на два изменения, означающее, что имеем два локальных коммита, которые не отправлены на сервер
 - ▶ `serverfix`-ветка отслеживает `server-fix-good`-ветку на нашем `teamone`-сервере и опережает на три коммита и отстает на один

Продолжение в следующей лекции...

