ТРПО, 11 неделя, лекция

Введение в работу с системой контроля версий Git

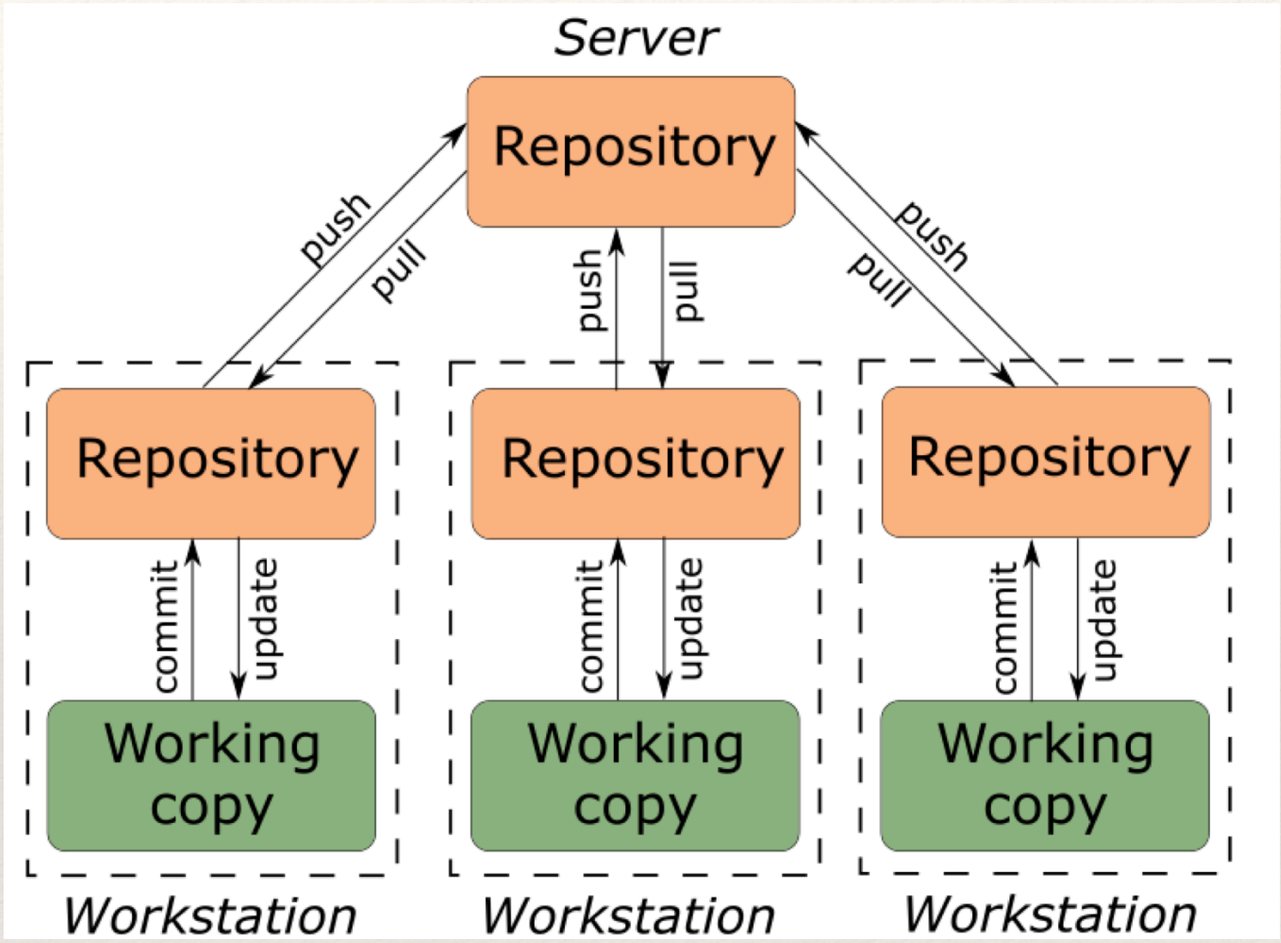
Основные понятия системы контроля версий

Системы контроля версий

- Система контроля версий (СКВ, Version Control System, VCS) - это программное обеспечение, которое позволяет отслеживать изменения в документах, фиксировать их, при необходимости производить откат
- Репозиторий (repository) - специальное хранилище папок проекта и файлов проекта, изменения в которых отслеживаются
- Рабочая копия (working copy) - это те файлы и папки, которые есть в распоряжении разработчика, с ними он непосредственно работает
- Commit - операция, фиксирующая измененияв рабочей копии проекта (отправляем изменения в репозиторий)

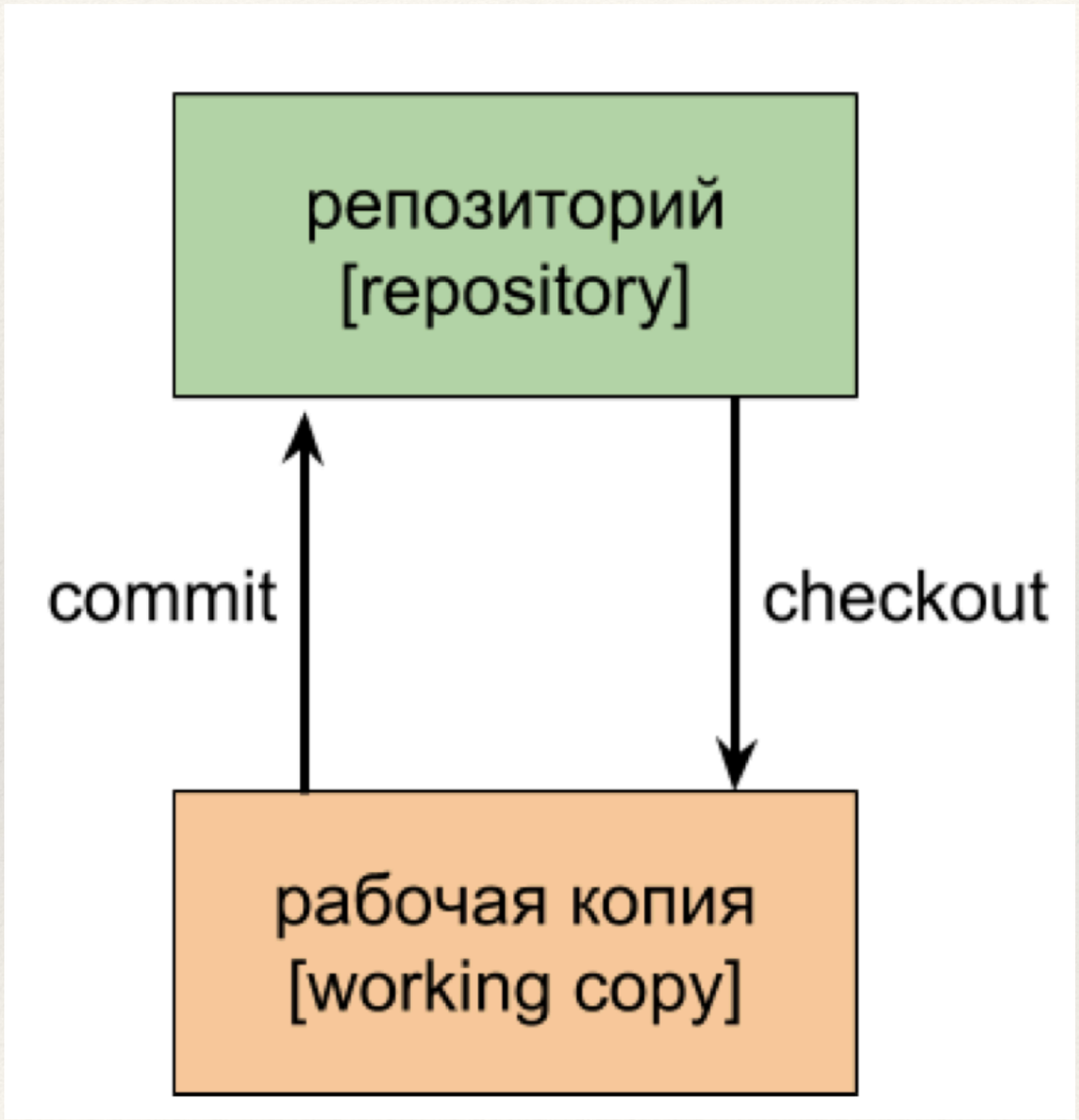
Распределенная система контроля версий

- Распределенная система контроля версий (Distributed Version Control System, DVSC) - позволяет хранить репозиторий (его копию) у каждого разработчика, можно и нужно синхронизировать локальные репозитории с удаленным
- Известные системы - Git, Mercurial…

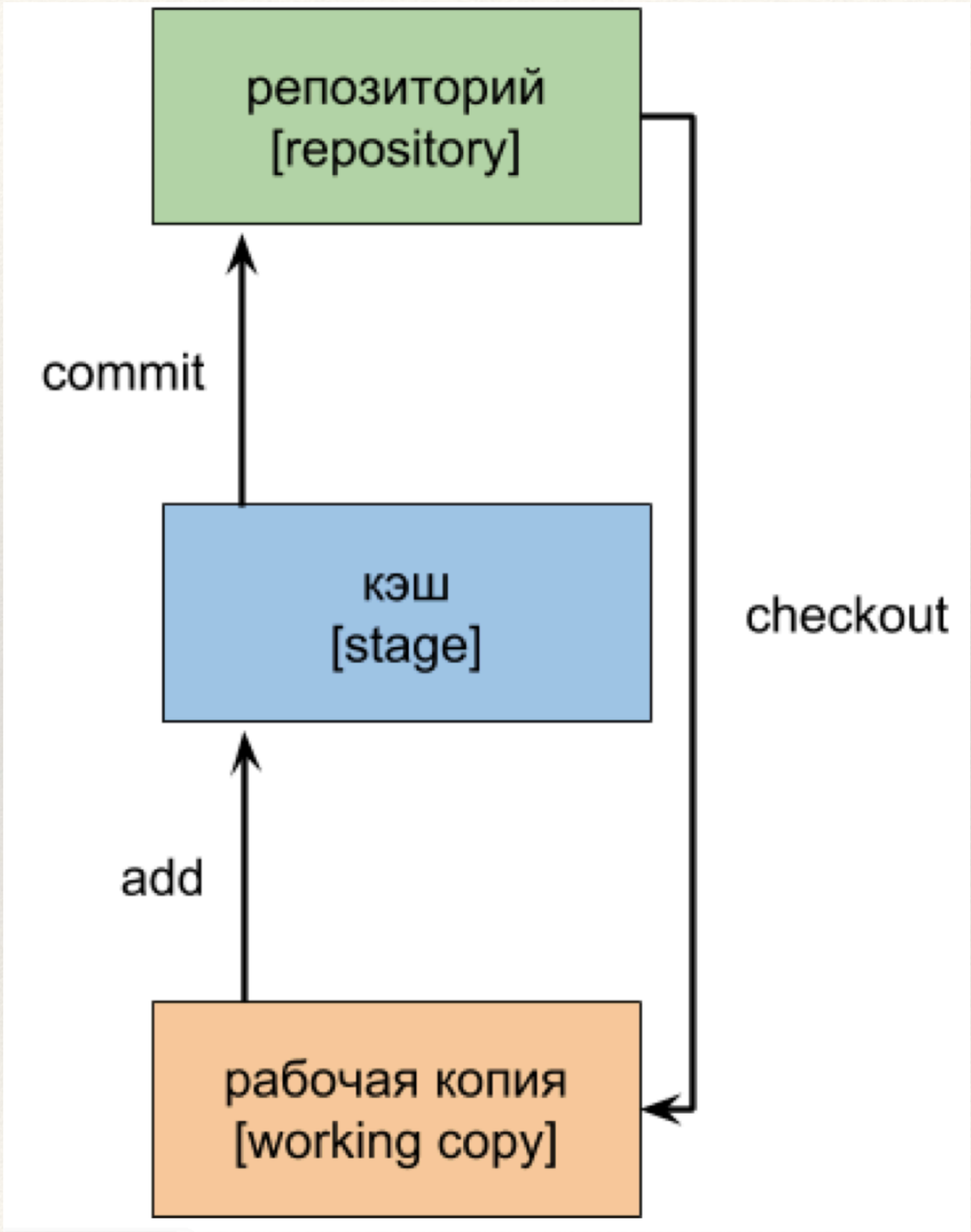


Пример локального и удаленного репозитория

Архитектура Git



Архитектура 2 деревьев



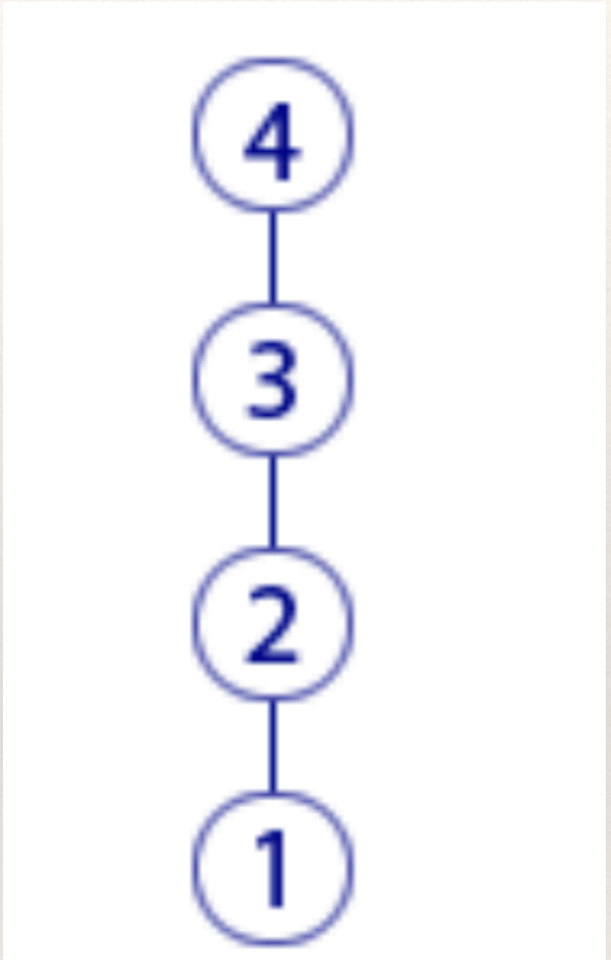
Архитектура 3 деревьев

Push, pull и разрешение конфликтов

- Прежде чем отправить изменения из локального репозитория на сервер (origin), т.е. прежде, чем сделать push, надо забрать с удаленного репозитория все проделанные на нем новые коммиты (pull)
- Если не делать так - возможен конфликт на стороне сервера (origin)
- Конфликт - ситуация, когда в одном и том же файле разные пользователи изменяют одни и те же строчки кода (или когда разные пользователи коммитят разные версиии бинарного файла)
- Разрешение конфликта - на стороне локального репозитория, можно либо выбрать один из предложенных вариантов, либо написать третий вариант, либо откатить изменения (abort)

Представление истории изменений

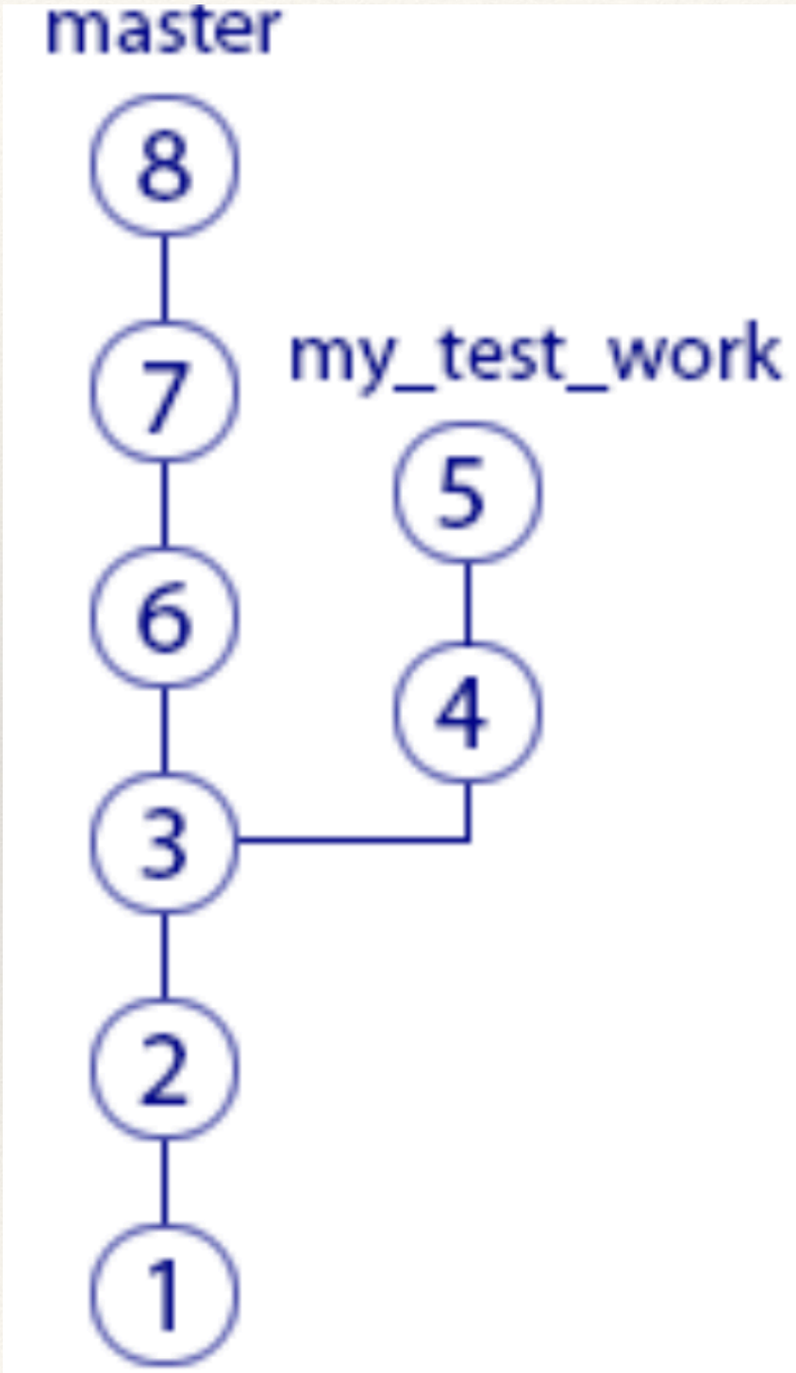
- История изменения - переход между коммитами
- Коммит тут можно рассматривать как
  - изменение файлов с предыдущего коммита;
  - слепок - состояние вообще всех файлов рабочего каталога после операции commit



Цепочка коммитов

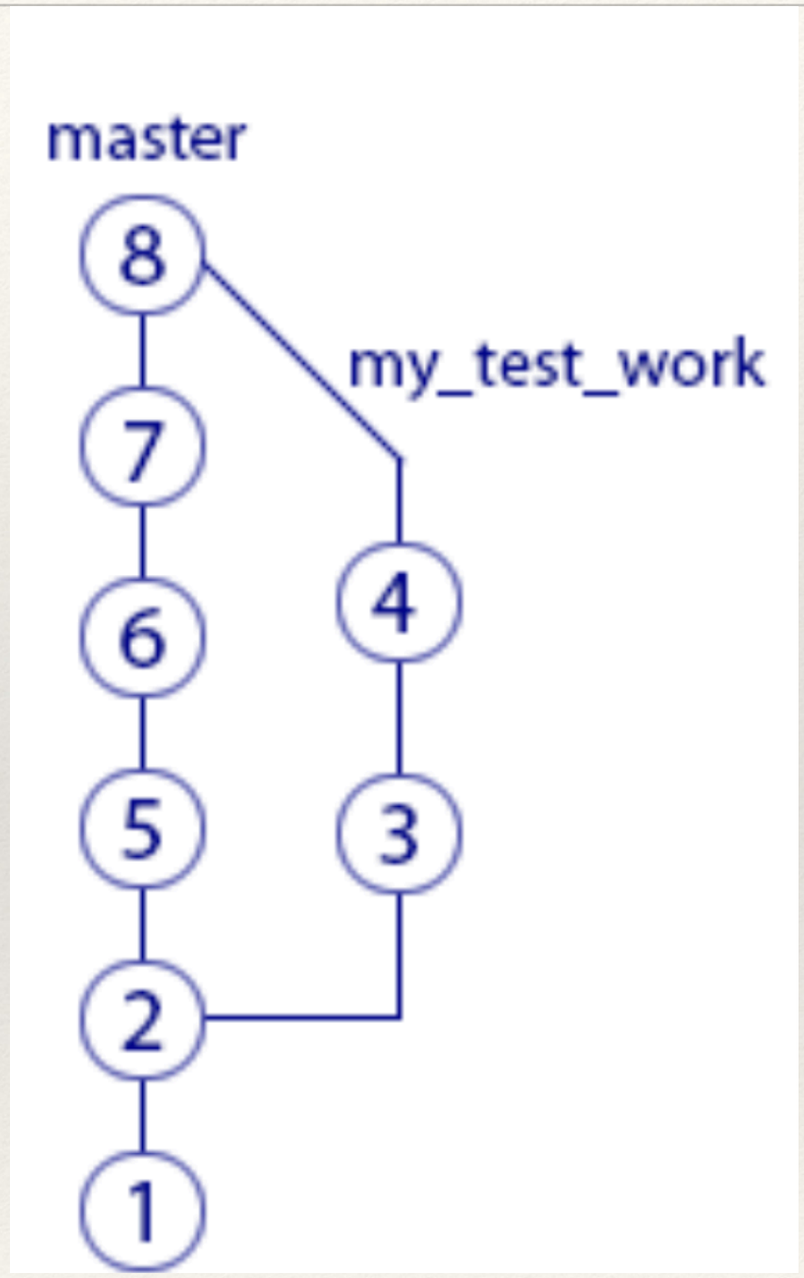
Концепция веток

- Ветвь нужна, чтобы в ней можно было проводить «независимую» работу, контролируемую системой версий, но при этом остальные разработчики не видят вашу историю работы в ветке
- У ветвей есть имена, по умолчанию работа ведется в ветке master
- Переключение между ветками - команда checkout



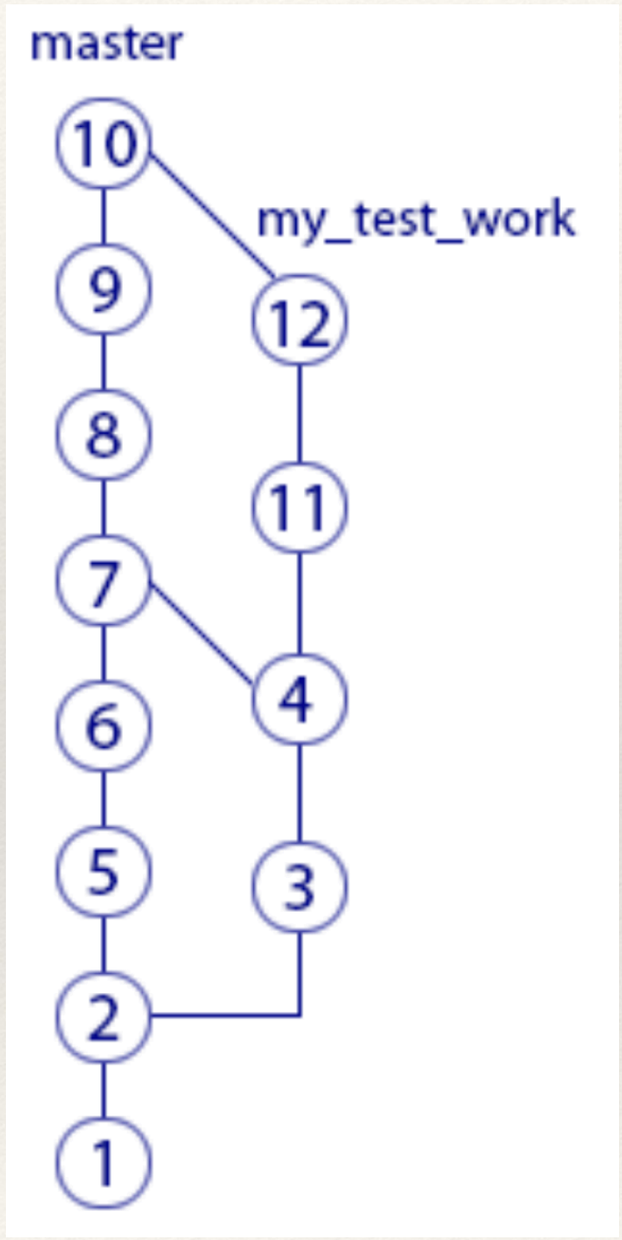
Слияние веток

- Слияние ветвей - merge
- Merge ничего не отправляет в удаленный репозиторий (origin), слияние производится в локальном репозитории
- Результат merge-commit надо отправлять (push) в удаленный репозиторий (origin)

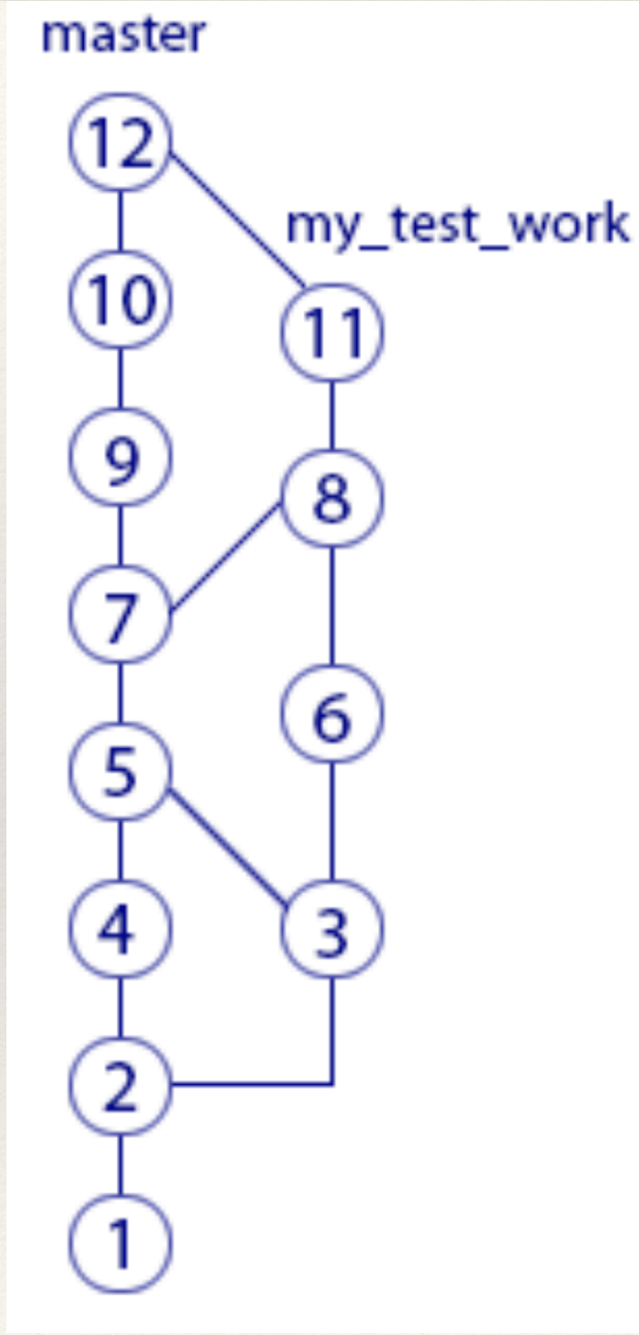


merge-commit

Множественные слияния веток



Несколько merge в основную ветку



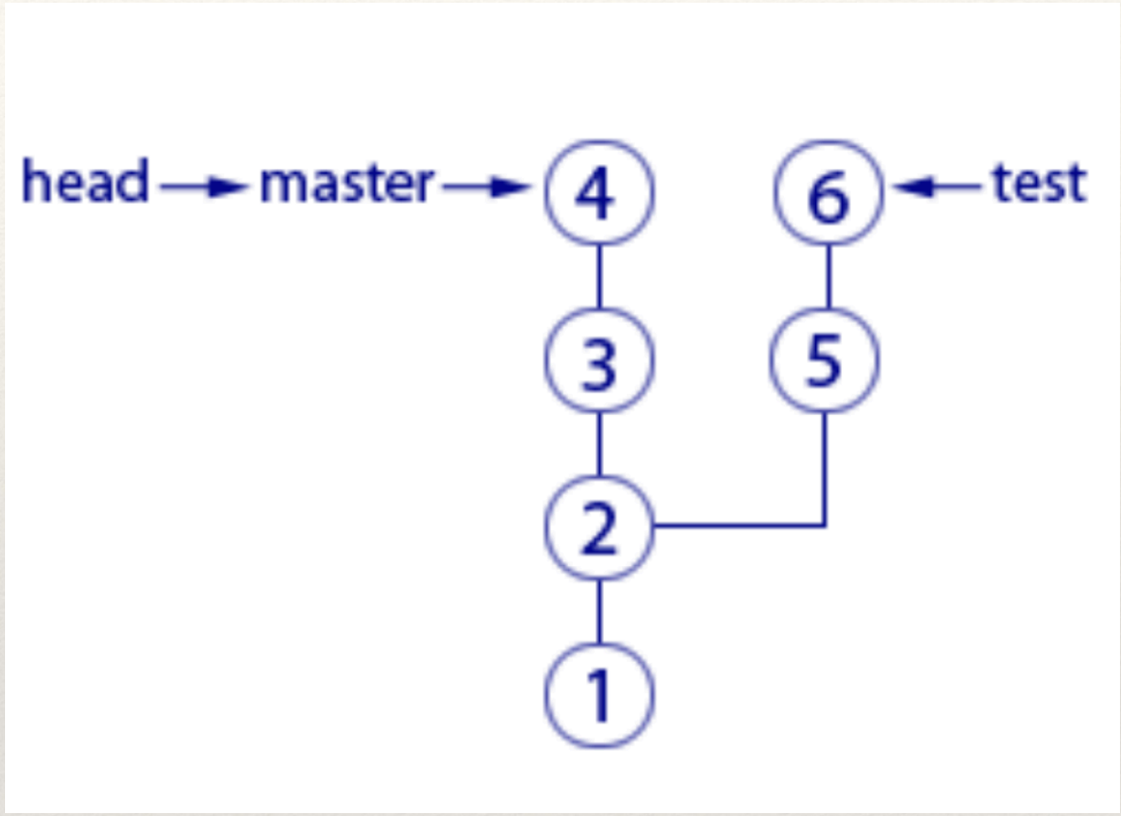
Несколько merge в обе стороны

Как Git различает коммиты и как различает ветки?

- Git каждому коммиту соответствует некоторый хэш e09844739f6f355e169f701a5b7ae02c214d5fb0
- Информацию о ветках Git хранит, используя концепцию указателей: имя_ветки-хэш_коммита master – e09844739f6f355e169f701a5b7ae02c214d5fb0
- Указатель ссылается на последний коммит ветки

Указатель head

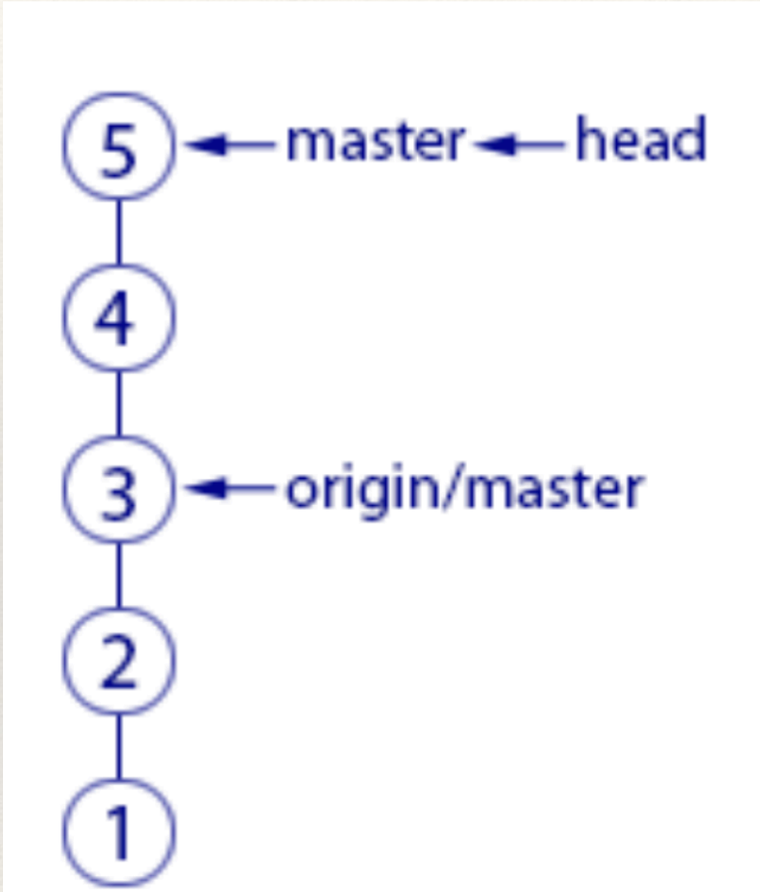
- Специальный указатель head смотрит на коммит, который выступает состоянием рабочего каталога
- head можно установить на любой коммит
- checkout переключит ветки, head автоматически будет смотреть на последний коммит в ветке



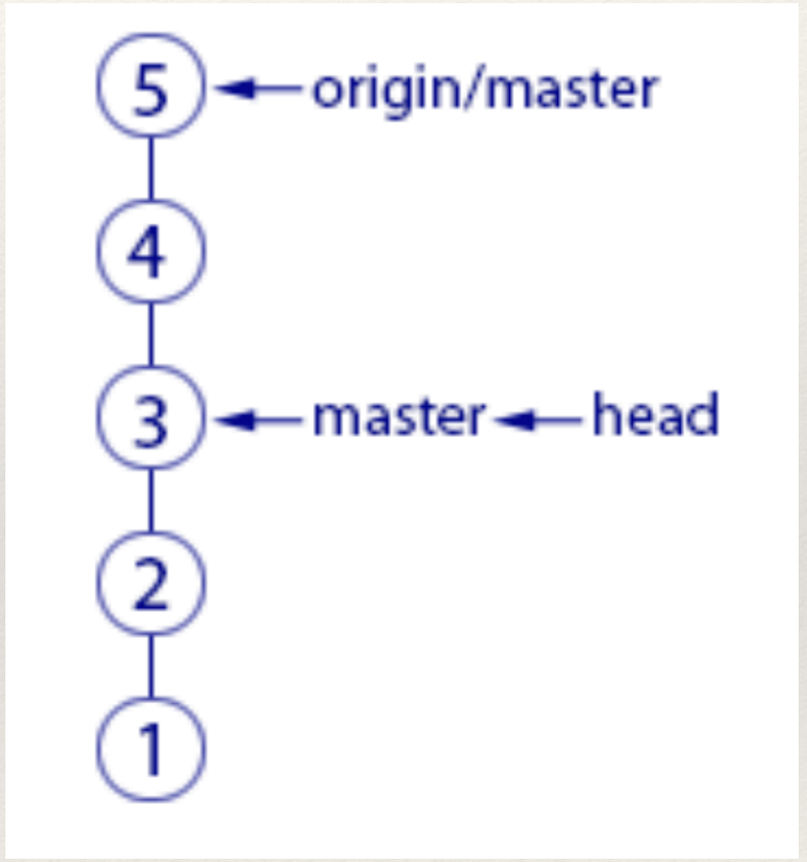
Указатель head

Указатель origin/master

- В удаленном репозитории тоже есть свои указатели веток
- pull забирает копию удаленного репозитория в локальный, а значит заберет из удаленного репозитория его указатель master
- master и origin/master могут смотреть на разные коммиты

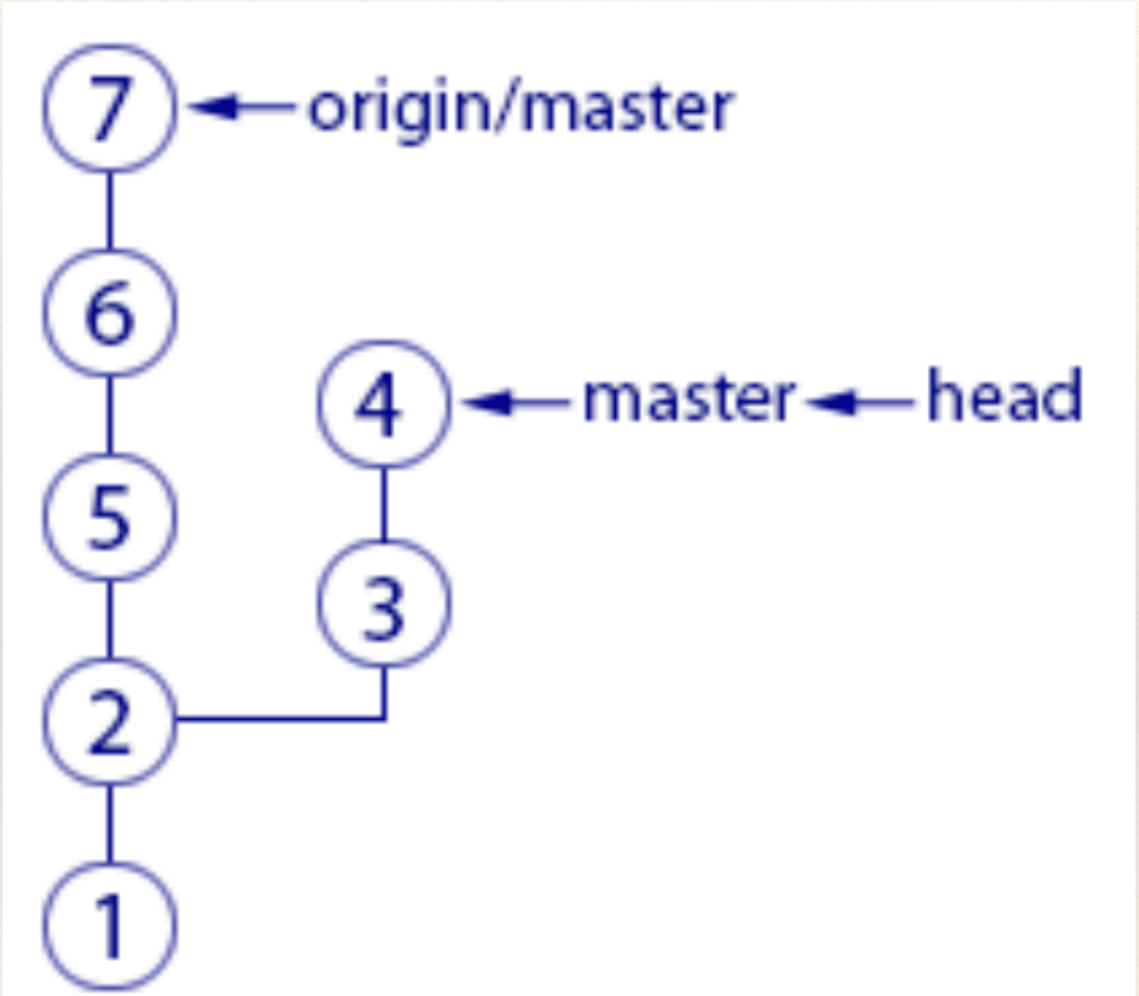


Указатель origin/master отстает

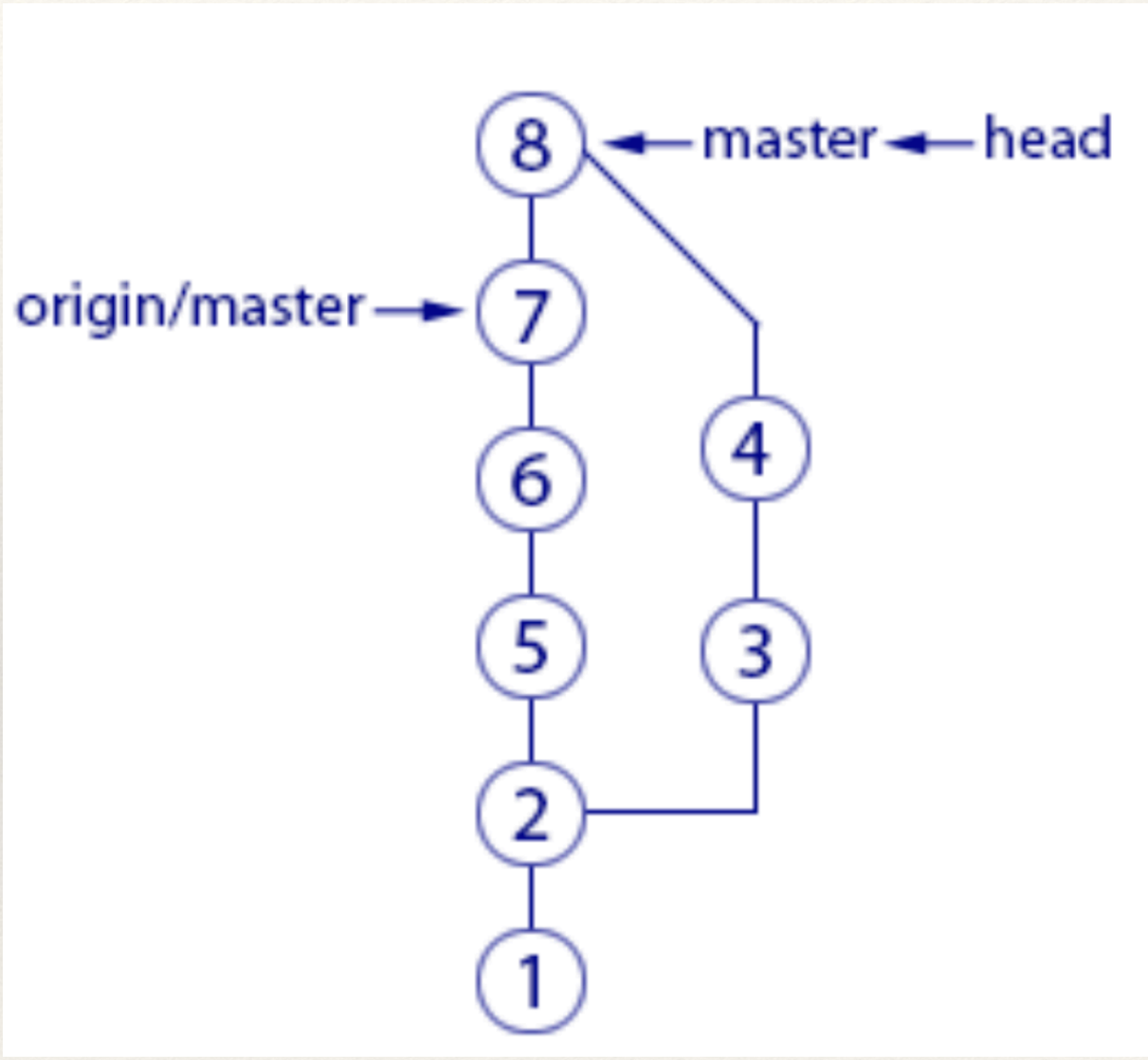


Указатель origin/master опережает (коммиты скачивали при помощи fetch, которая забирает коммиты, но не обновляет рабочий каталог)

Fetch и ветви

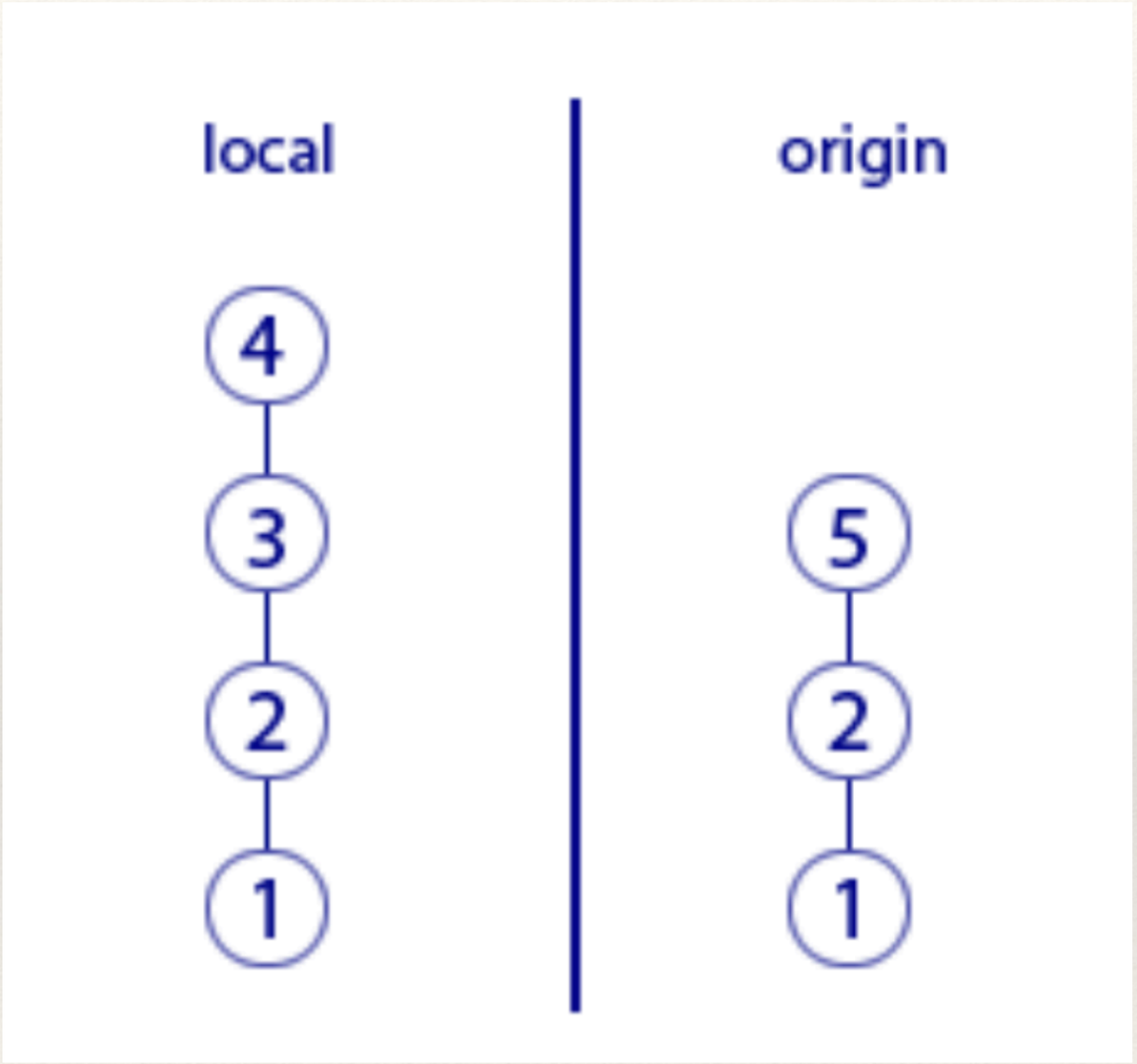


Ветка в результате fetch



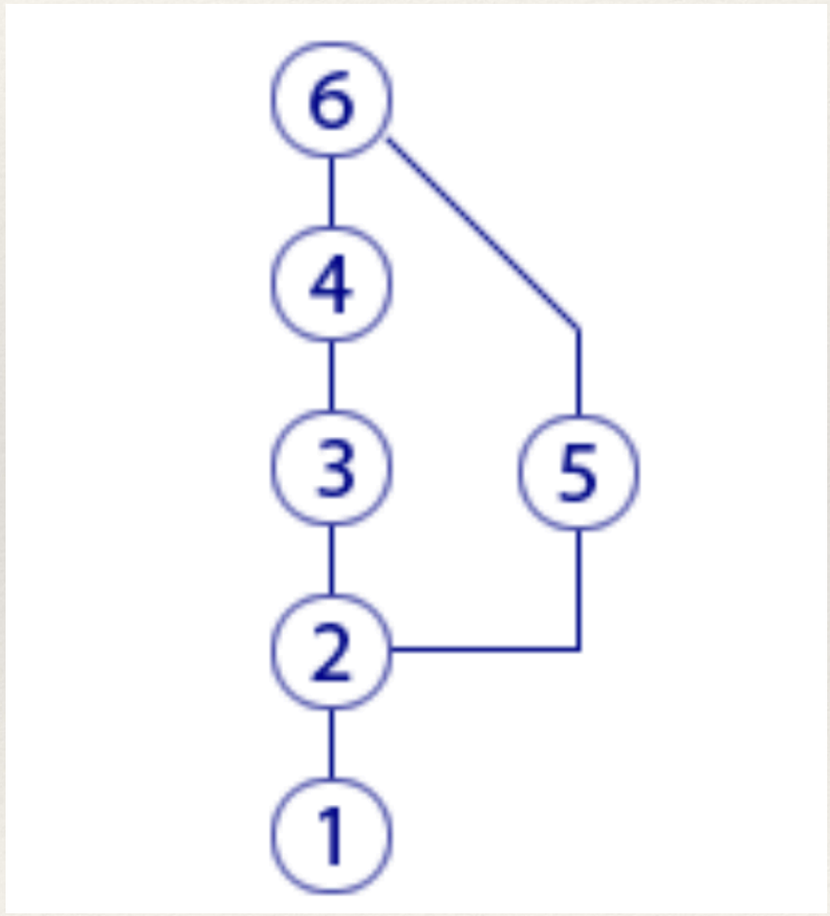
После merge

Push и ошибки



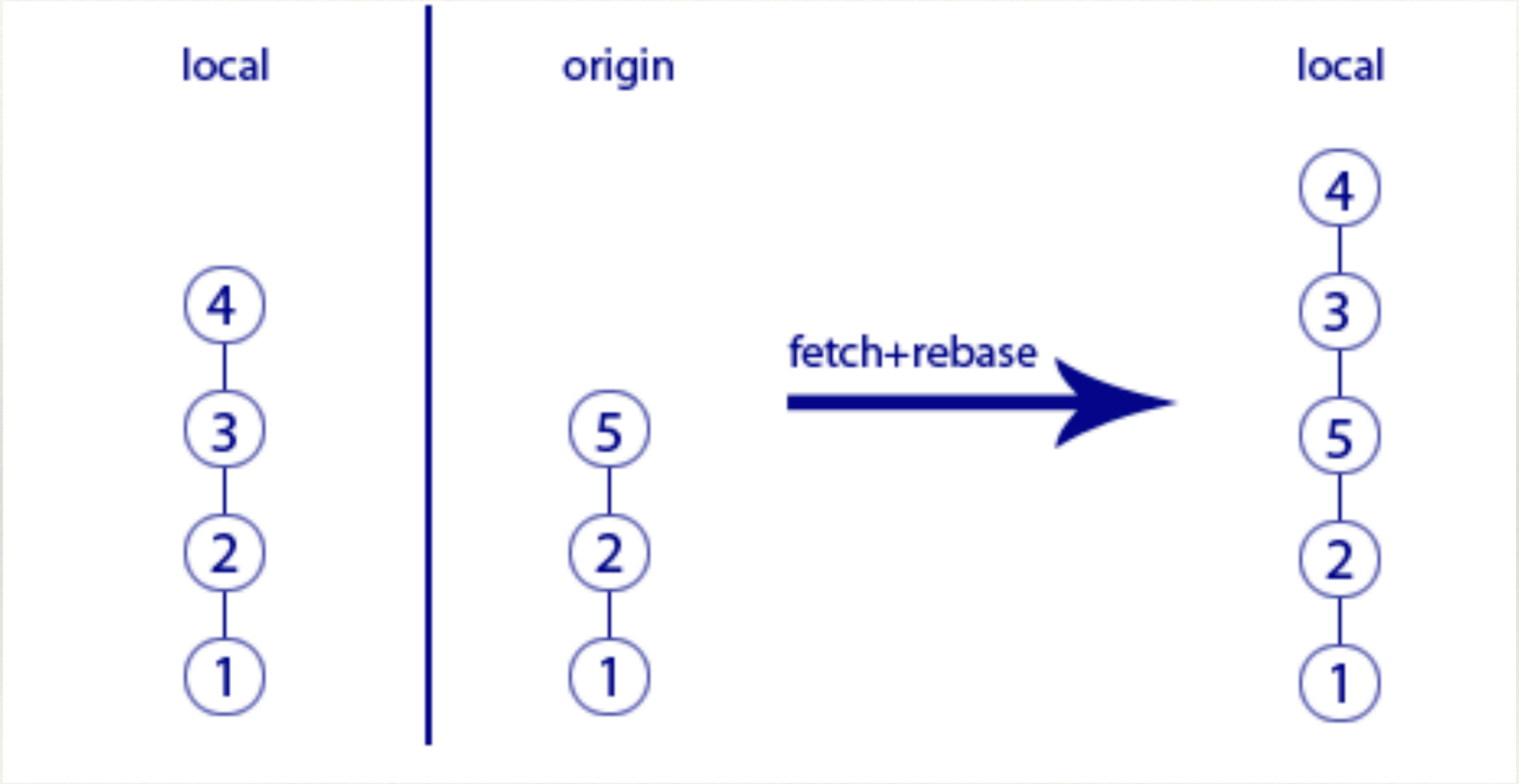
Сначала коммиты 1 и 2 скачаны в локальный репозиторий, потом вы делаете коммиты 3 и 4 в локьном, кто-то делает коммит 5 и отправляет его на сервер

Попытка push даст ошибку - т.к. коммиты 3 и 4 надо цеплять ко 2, а там уже есть 5



pull-merge-push

Rebase



rebase

Для дальнейшей работы

Работа с Гит

http://bioinformaticsinstitute.ru/sites/default/files/instrukciya_po_ispolzovaniyu_git_dlya_nachinayushchih.pdf

Теория:

Официальный учебник Git http://git-scm.com/book/ru/v2

Просто и коротко о концепции Git:

https://habr.com/ru/company/playrix/blog/345732/

https://habr.com/ru/company/playrix/blog/348864/

https://habr.com/ru/company/playrix/blog/350492/

Еще блог (более сложно) https://habr.com/ru/company/intel/blog/344962/

Для дальнейшей работы

Практика:

Еще пример работы с Git

http://asu.ugatu.ac.ru/library/101/767b2bbfc9aabcb861809924a897df03.pdf

Интерактивное обучение Git https://try.github.io

Курс на степике по Git https://stepik.org/course/3145/syllabus

GUI-клиенты Git для разработчиков

https://techrocks.ru/2020/04/24/best-git-gui-for-mac-linux-windows/