



Основы git



План

1. Системы контроля версий

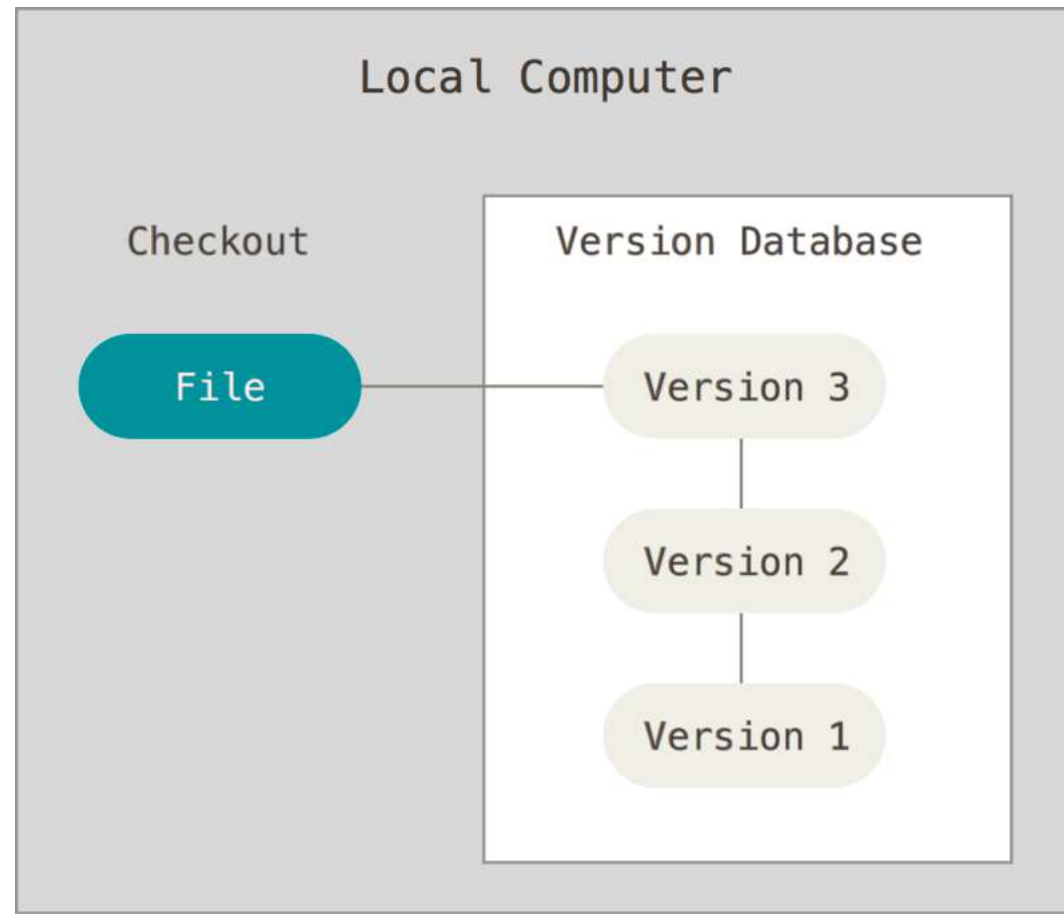
2. Git. Основные команды

3. Git. Продвинутое использование

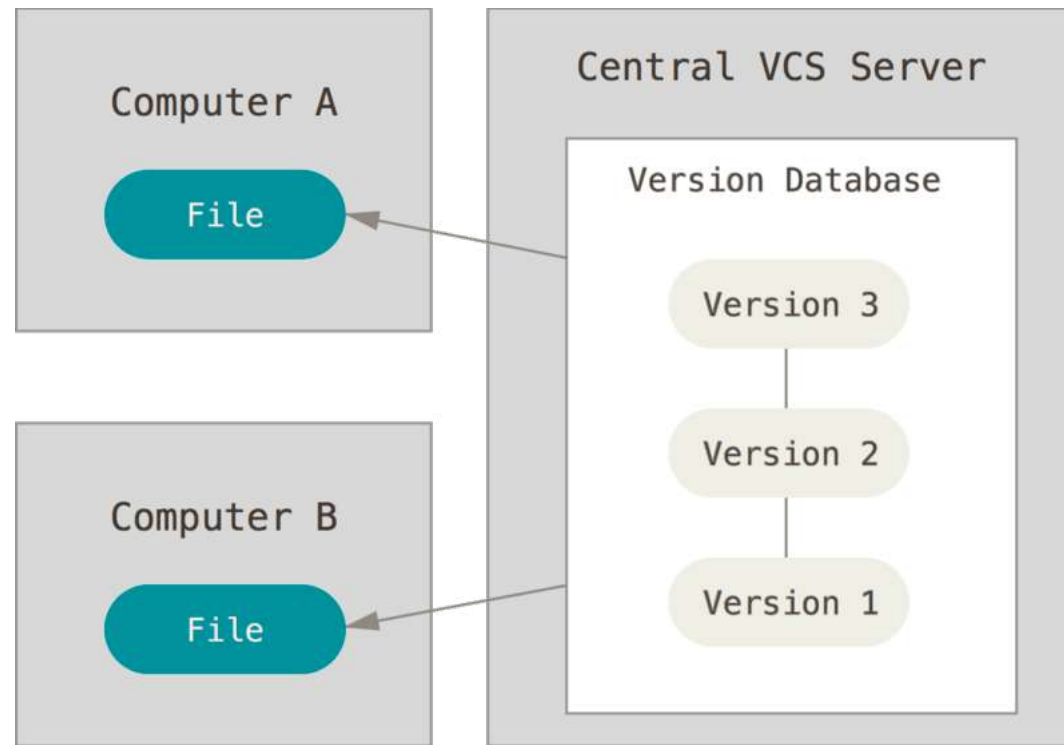
1. Системы контроля версий



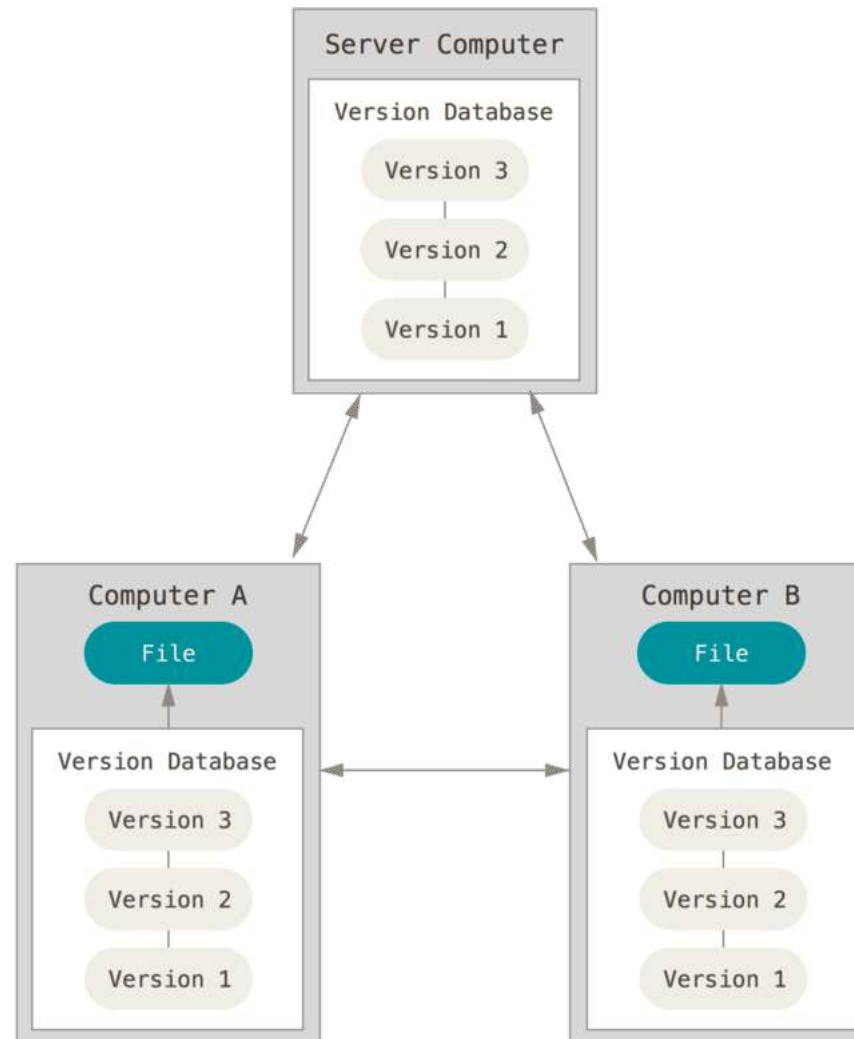
Локальные системы контроля версий



Централизованные системы контроля версий



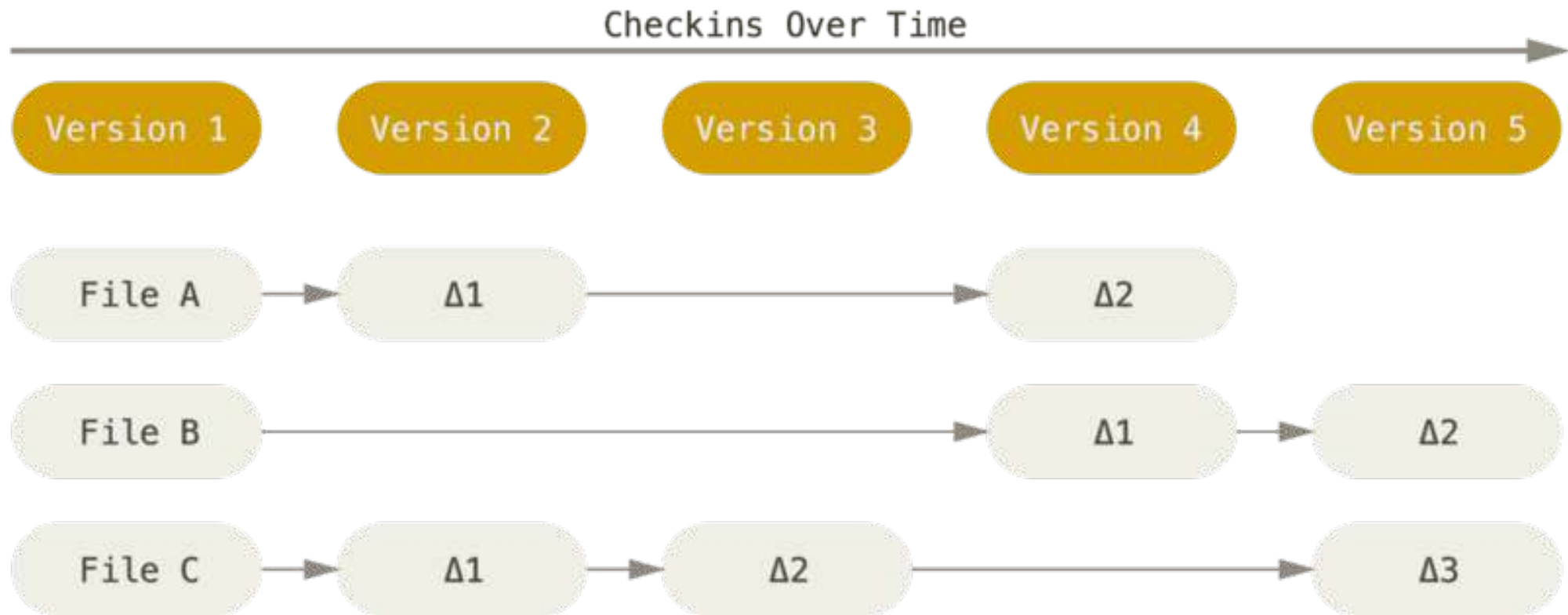
Распределённые системы контроля версий



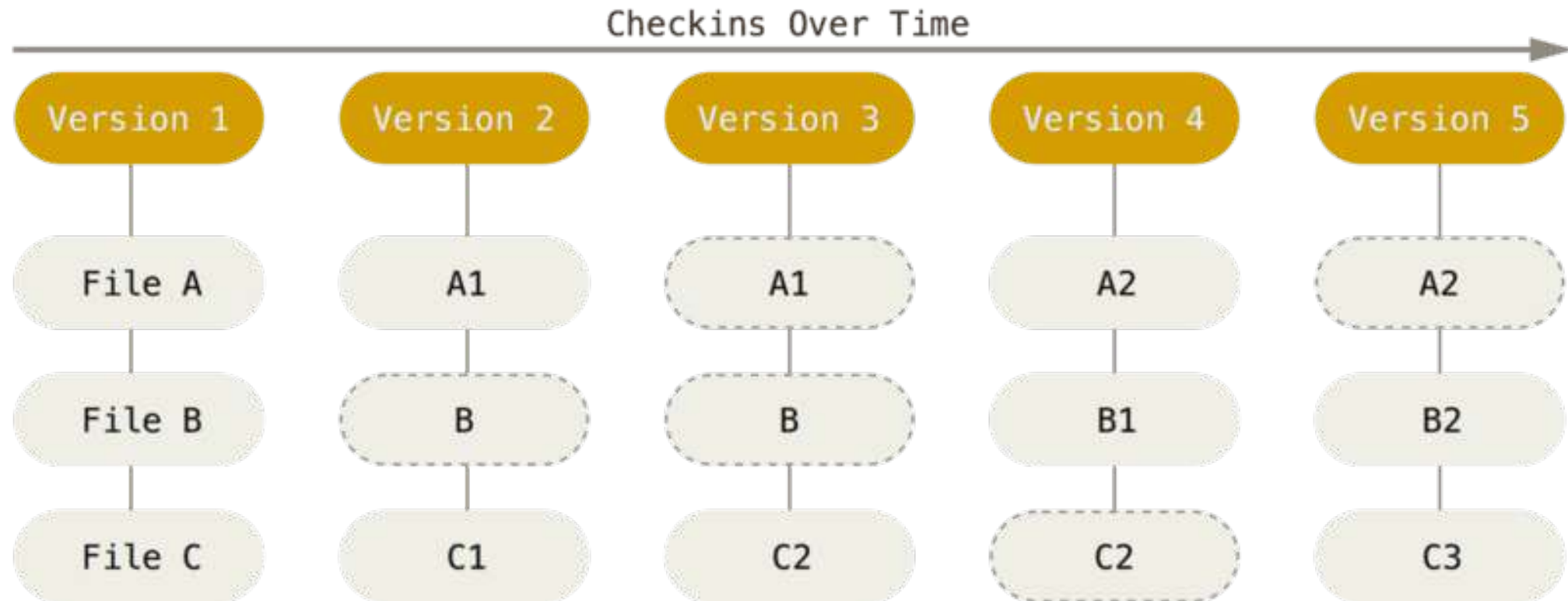
Git

- Скорость
- Простая архитектура
- Хорошая поддержка нелинейной разработки (тысячи параллельных веток)
- Полная децентрализация
- Возможность эффективного управления большими проектами, такими как ядро Linux (скорость работы и разумное использование дискового пространства)

Хранение изменений

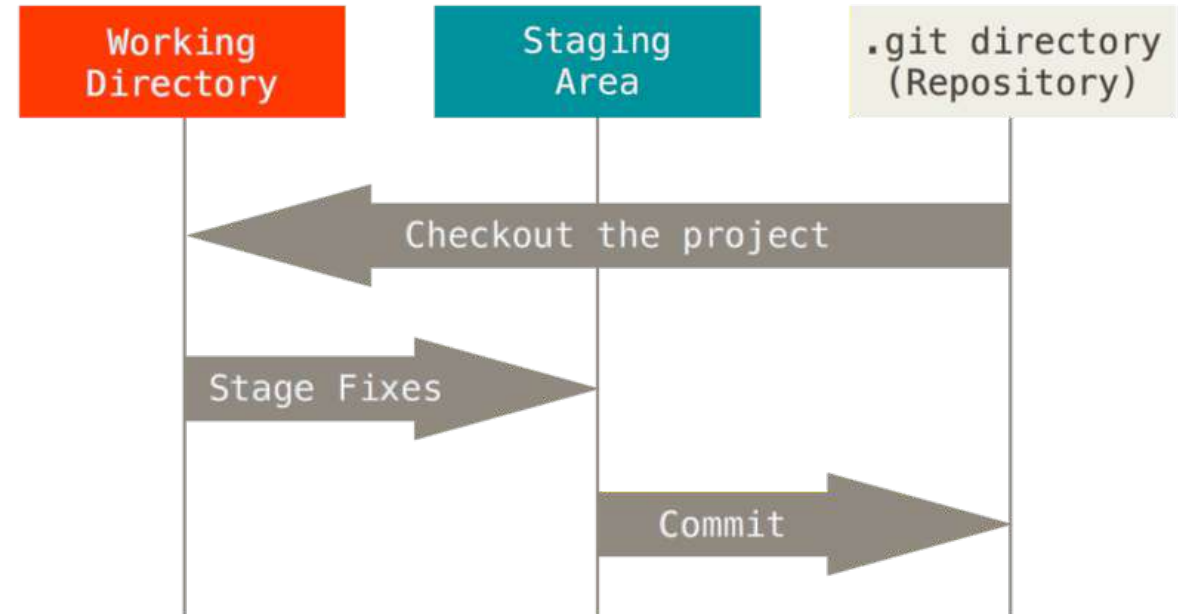


Хранение снимков

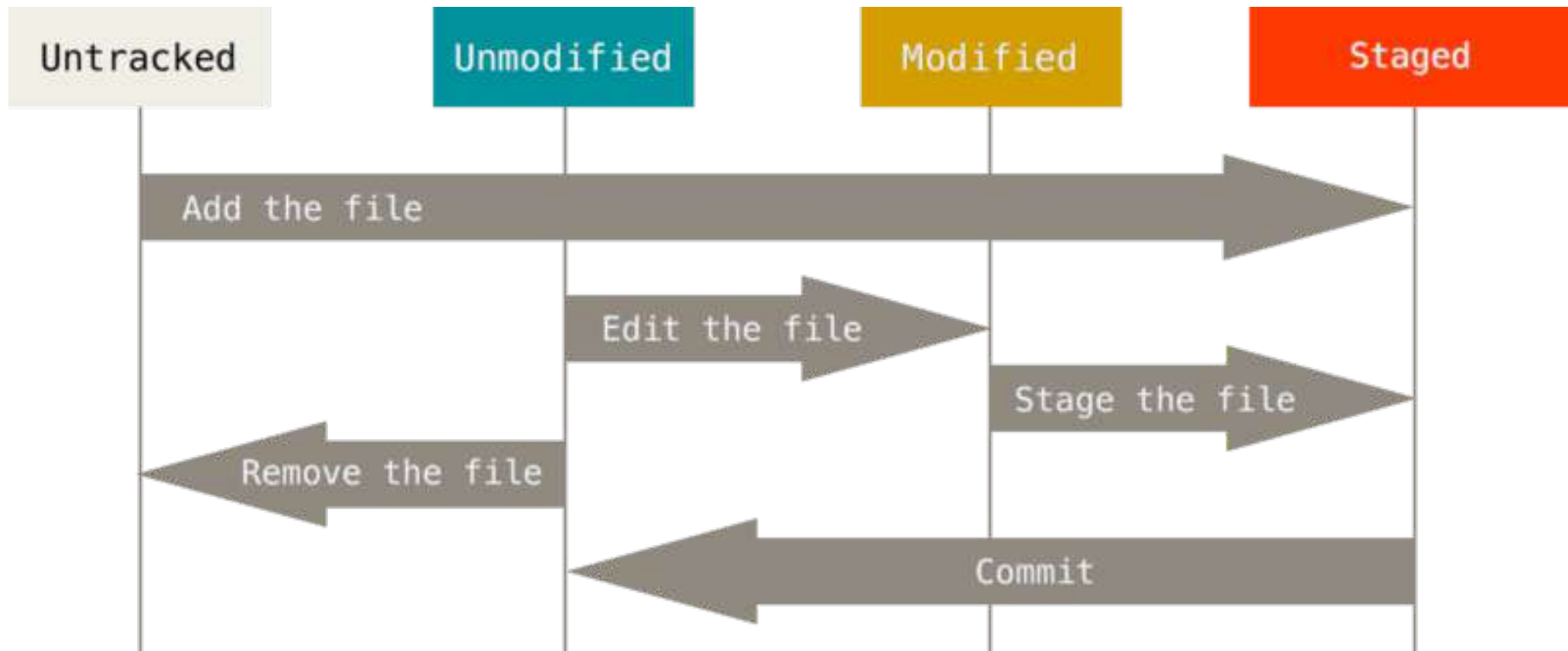


Основные состояния файлов

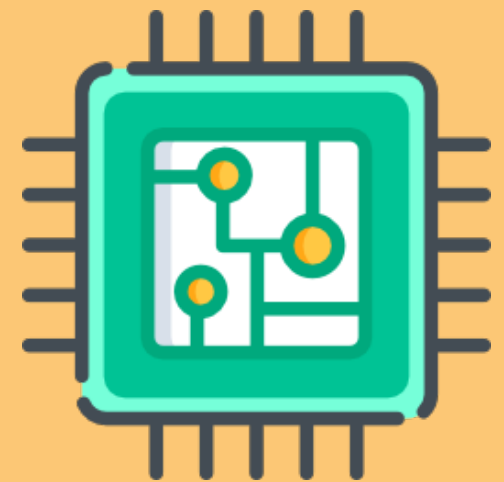
- зафиксированное (committed)
- изменённое (modified)
- подготовленное (staged)



Жизненный путь файла



2. Git. Основные команды



Основные команды

1. `git clone` – клонирование репозитория
2. `git status` – получение информации по репозиторию
3. `git add` – индексирование файла
4. `git diff` – изменения между текущим состоянием и последним проиндексированным
5. `git commit` – добавление stage в индекс
6. `git rm` – удаление файла

Работа с историей состояний

1. `git log` – история коммитов
2. `git commit –amend` – редактирование последнего коммита
3. `git reset HEAD` – откатывание файлов к последнему коммиту
4. `git checkout <filepath>` – откатывание состояния к последнему коммиту для конкретного файла

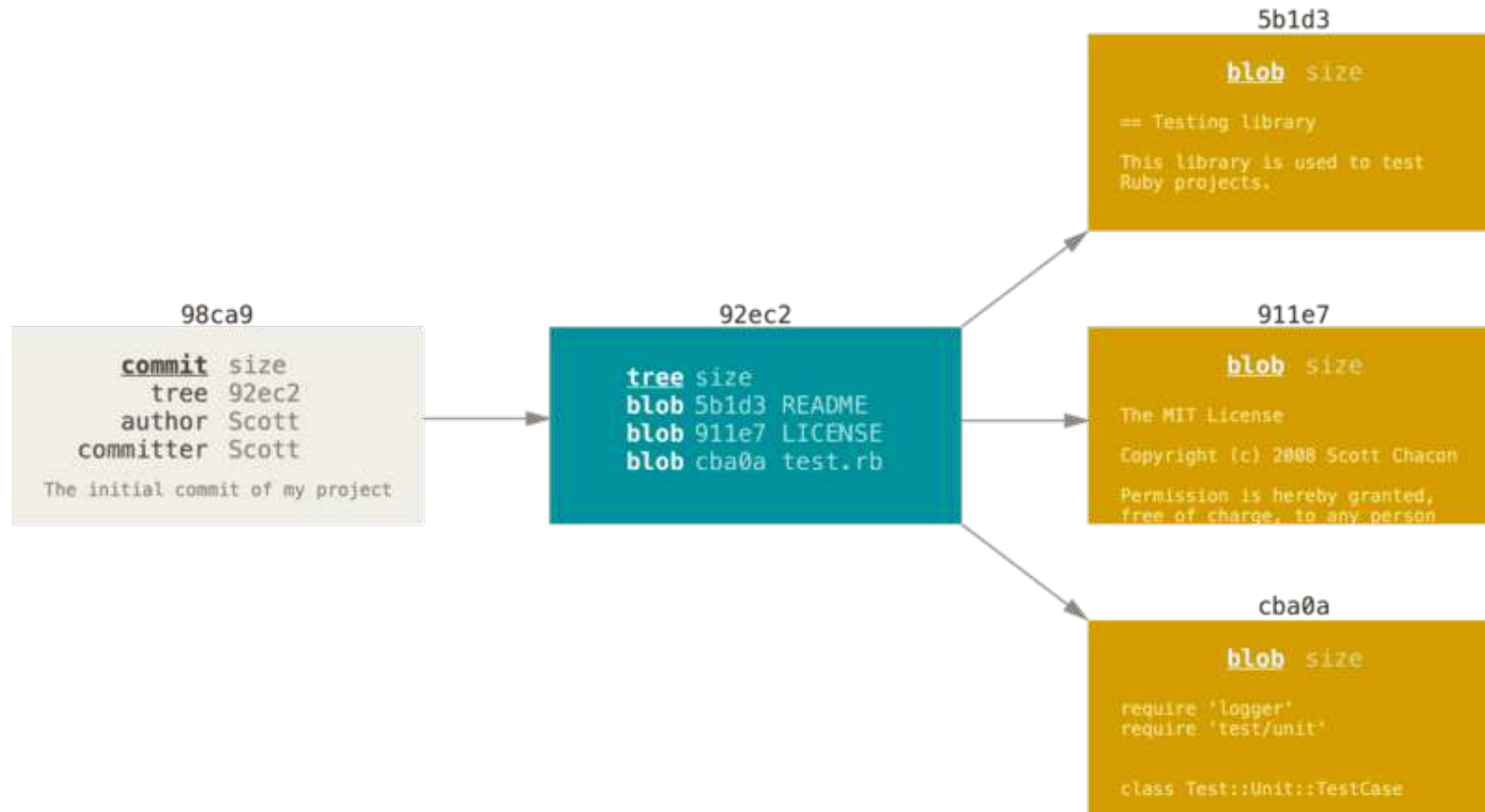
Работа с удалёнными серверами

1. `git fetch` – получить изменения с удалённого репозитория
2. `git pull` – получить и применить изменения с удалённого репозитория
3. `git push` – отправить изменения на удалённый репозиторий

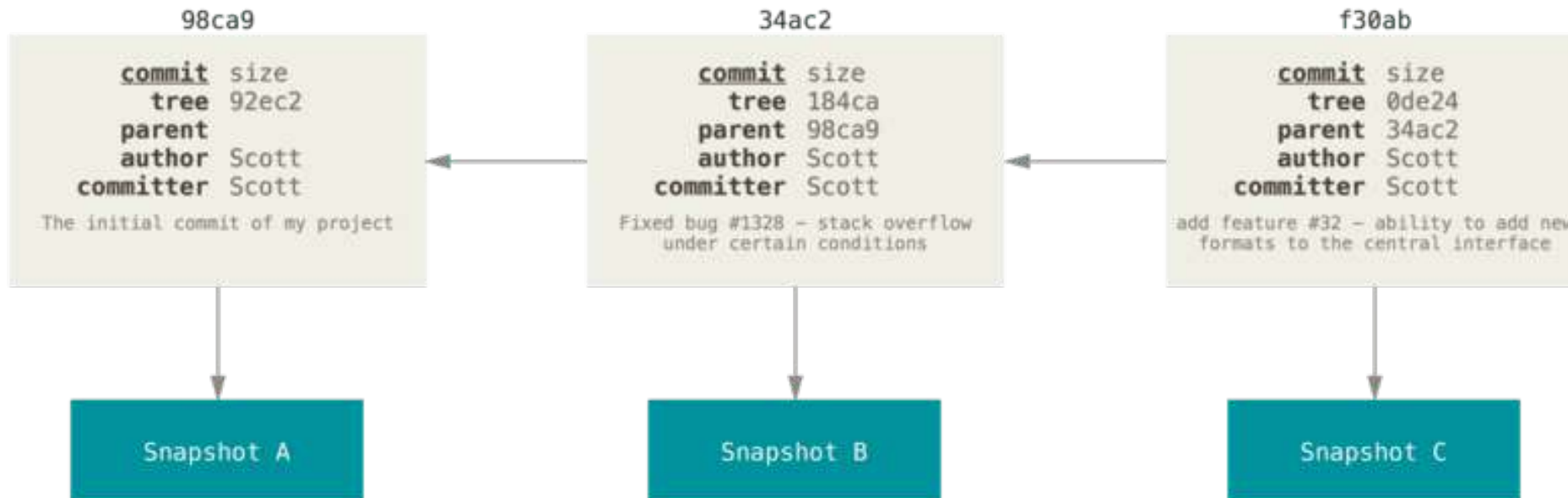
3. Git. Продвинутое использование



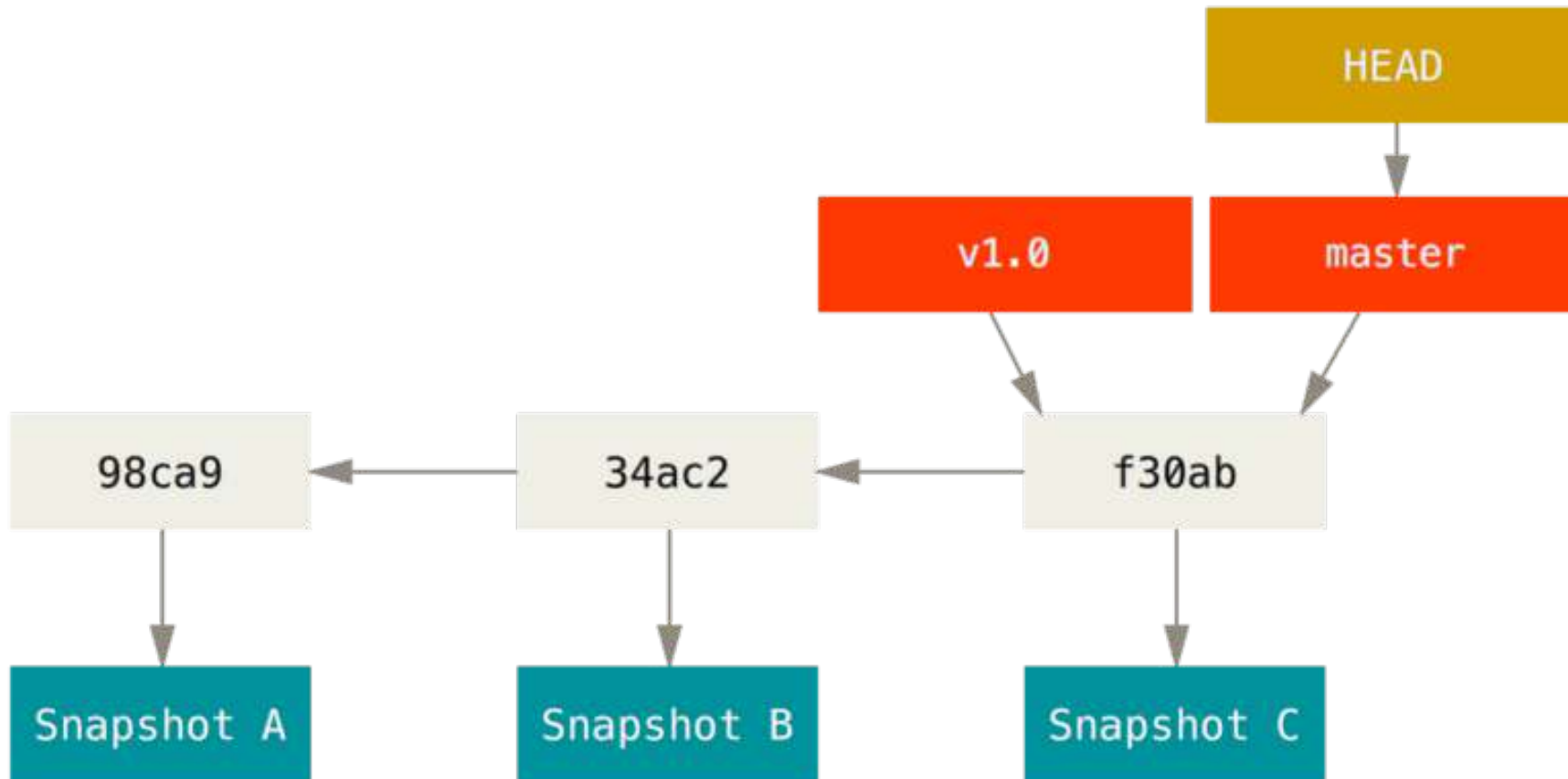
Содержание коммита



Несколько КОММИТОВ

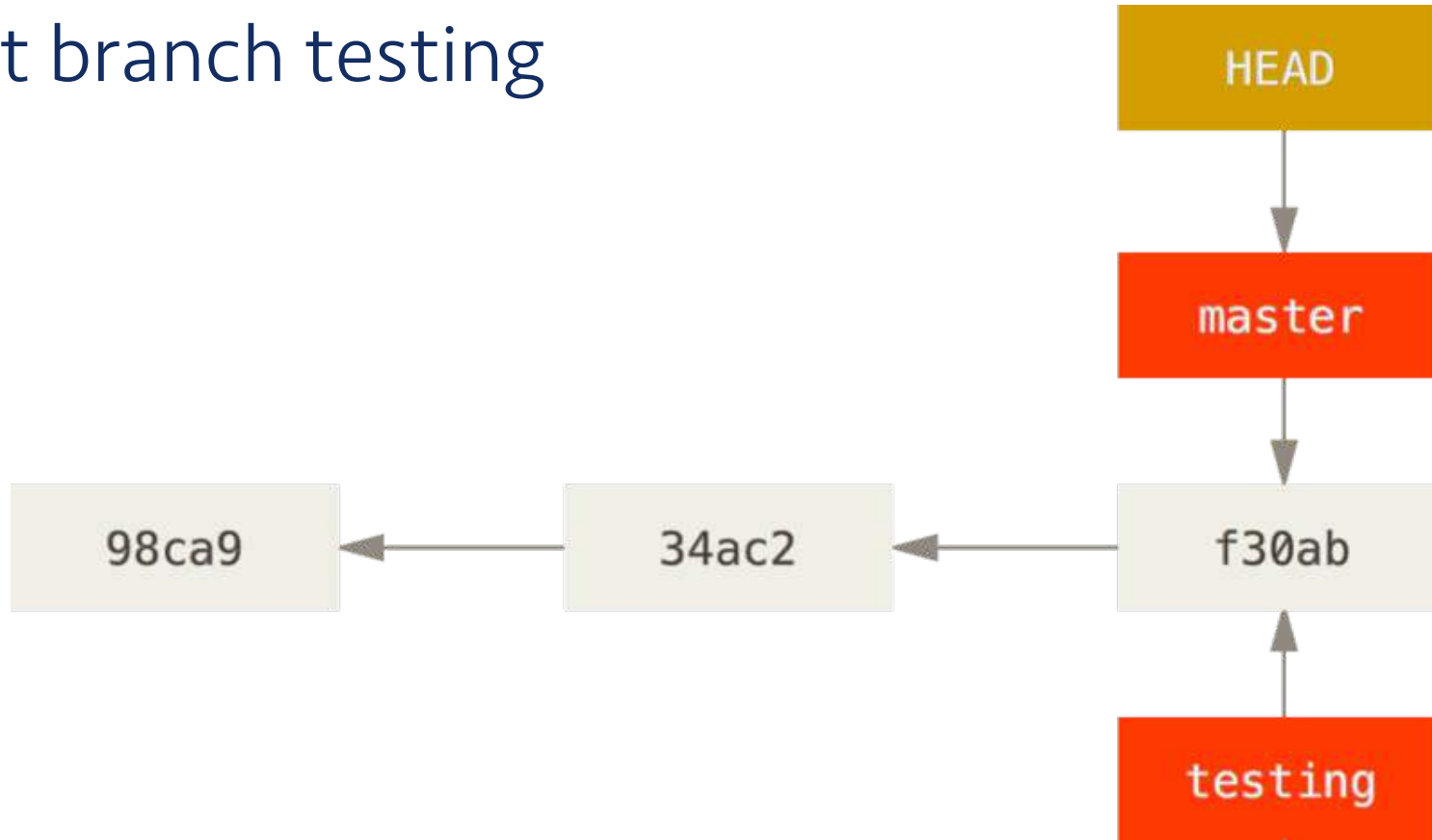


Ветвление в Git



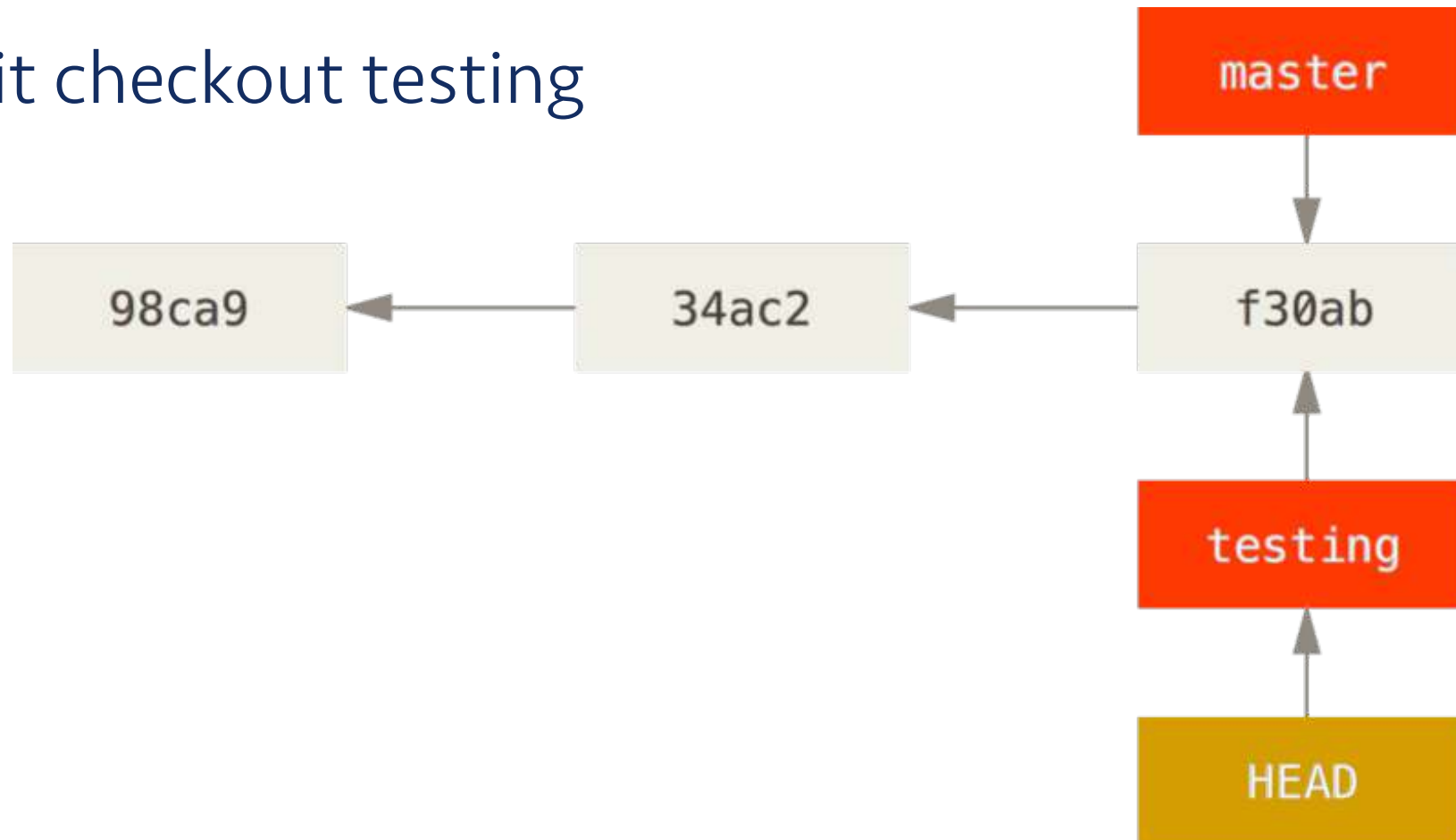
Создание новой ветки

- git branch testing



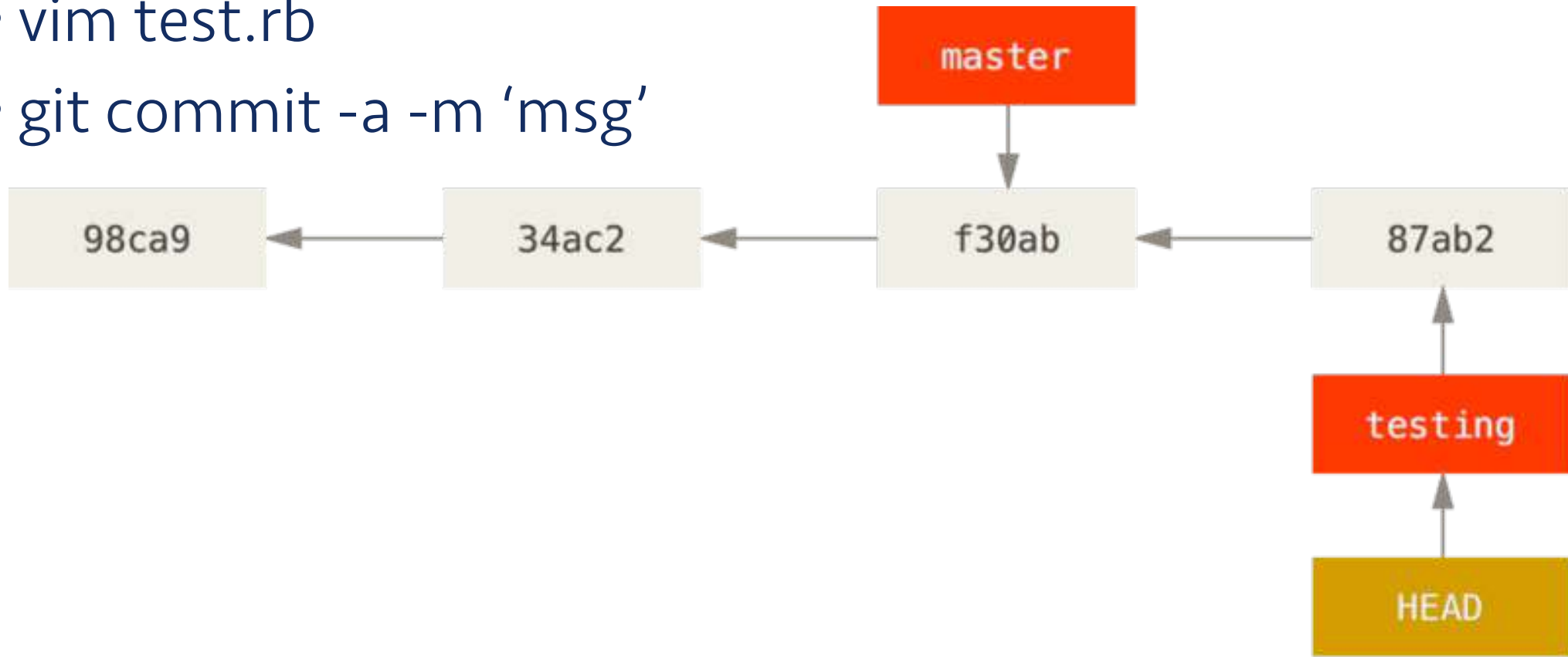
Переключение веток

- git checkout testing



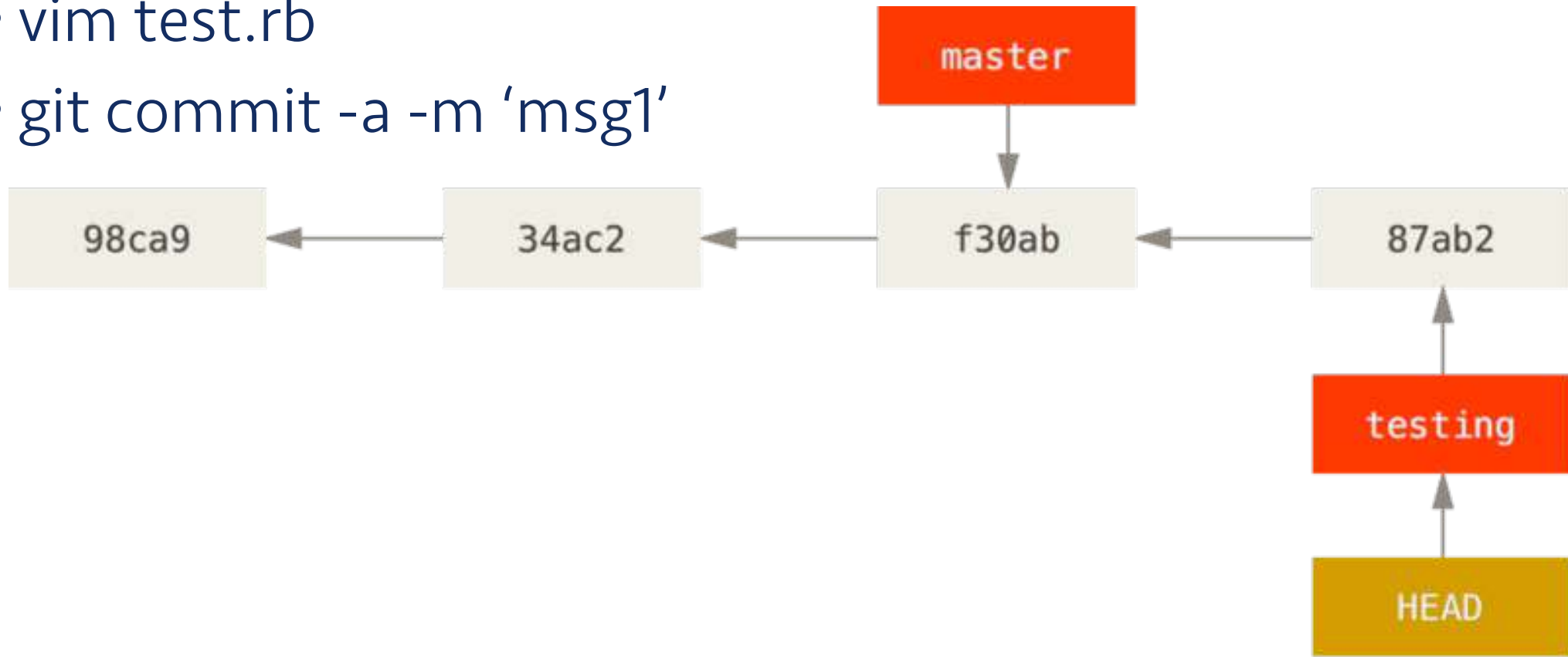
Переключение веток

- `vim test.rb`
- `git commit -a -m 'msg'`



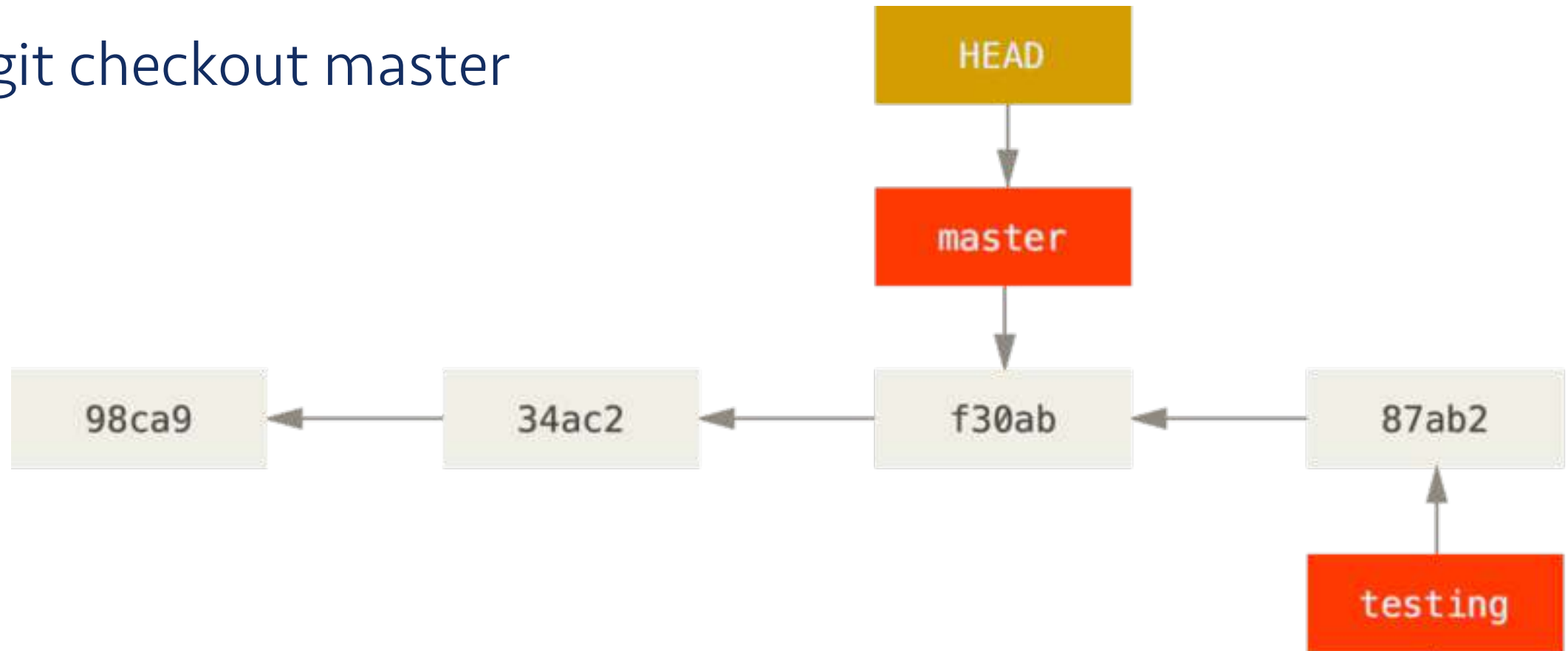
Переключение веток

- `vim test.rb`
- `git commit -a -m 'msg1'`



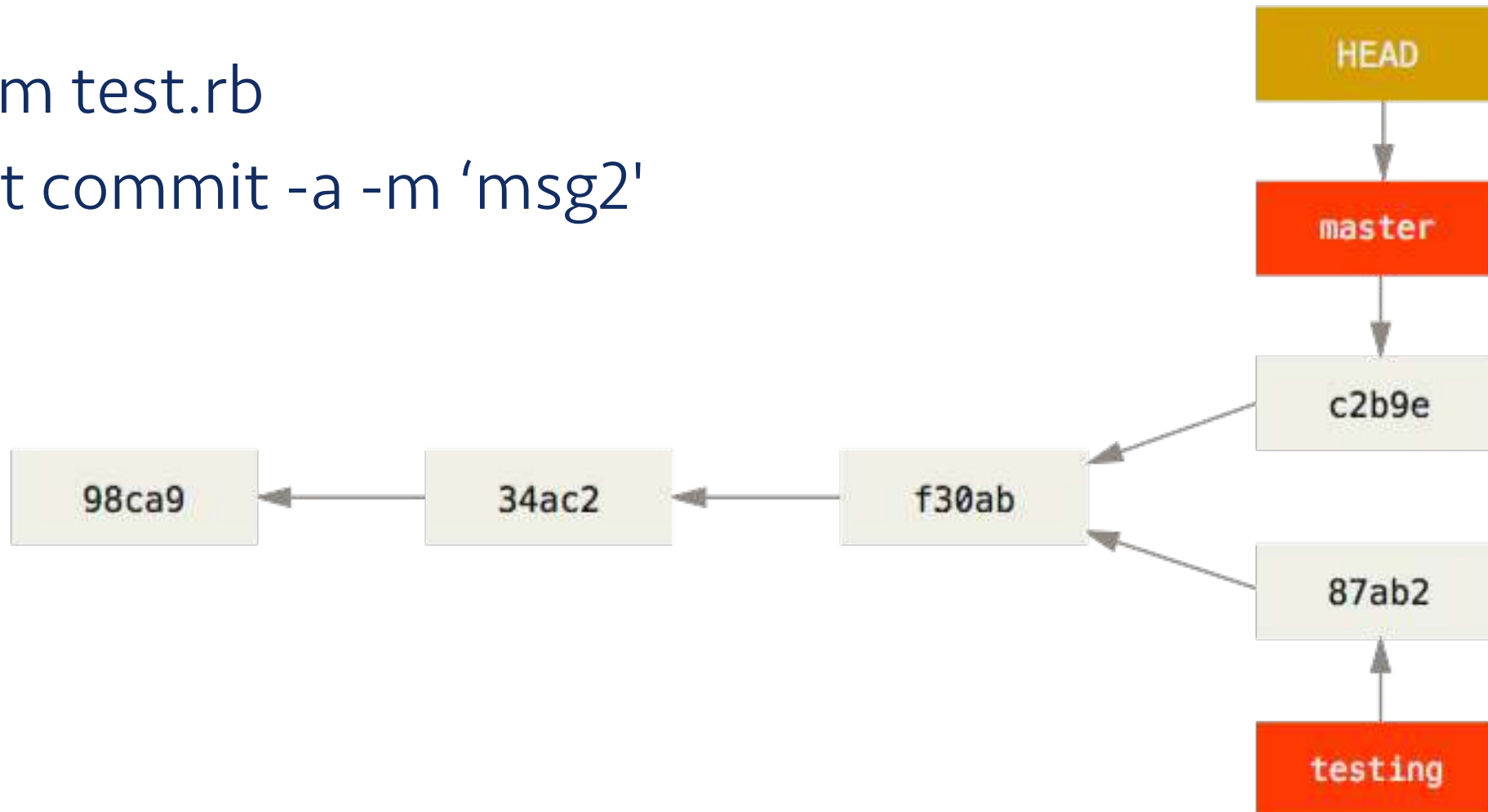
Переключение веток

- `git checkout master`

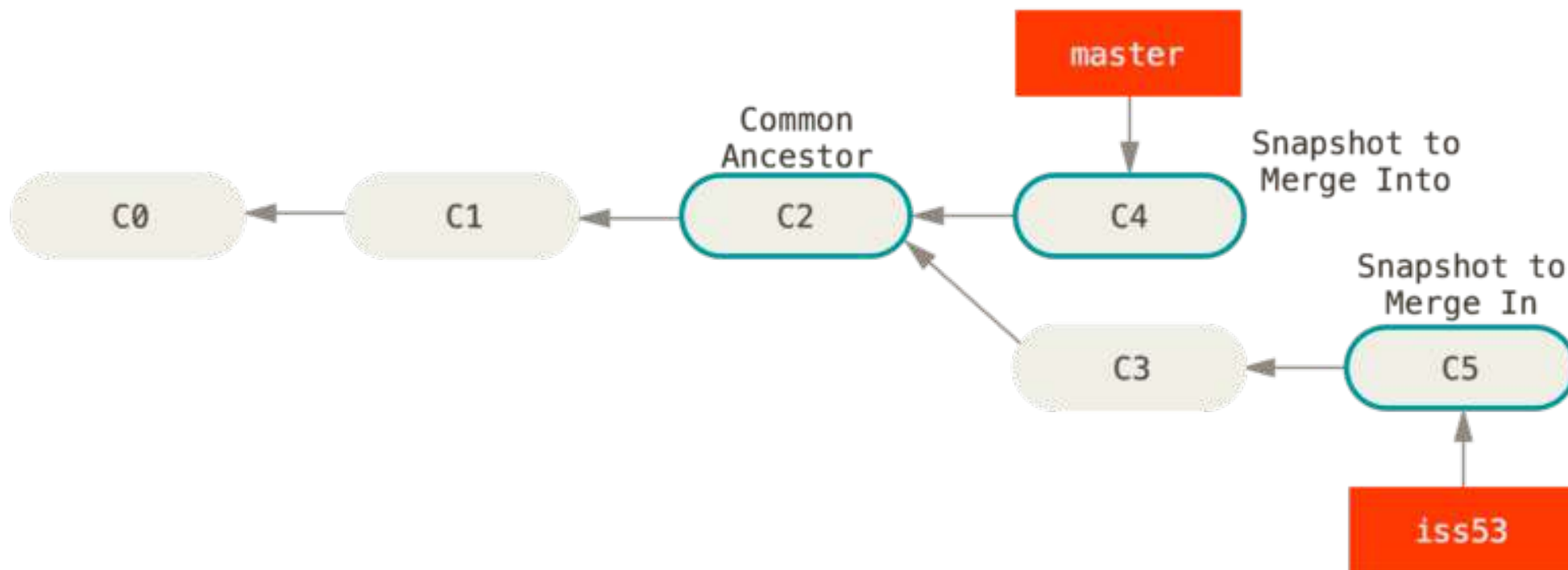


Переключение веток

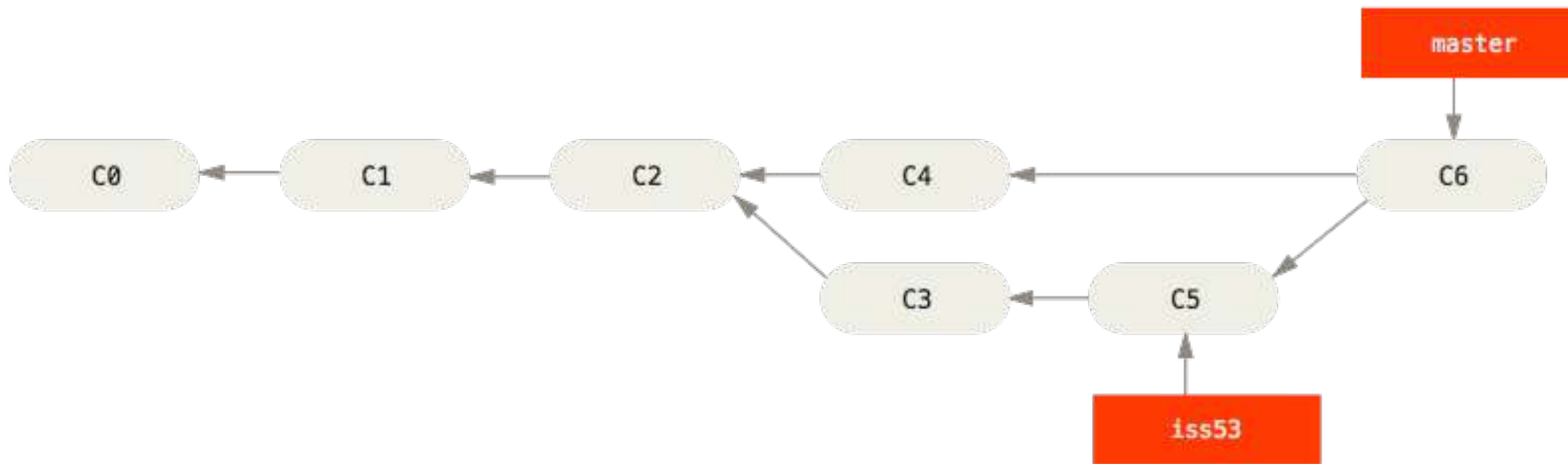
- `vim test.rb`
- `git commit -a -m 'msg2'`



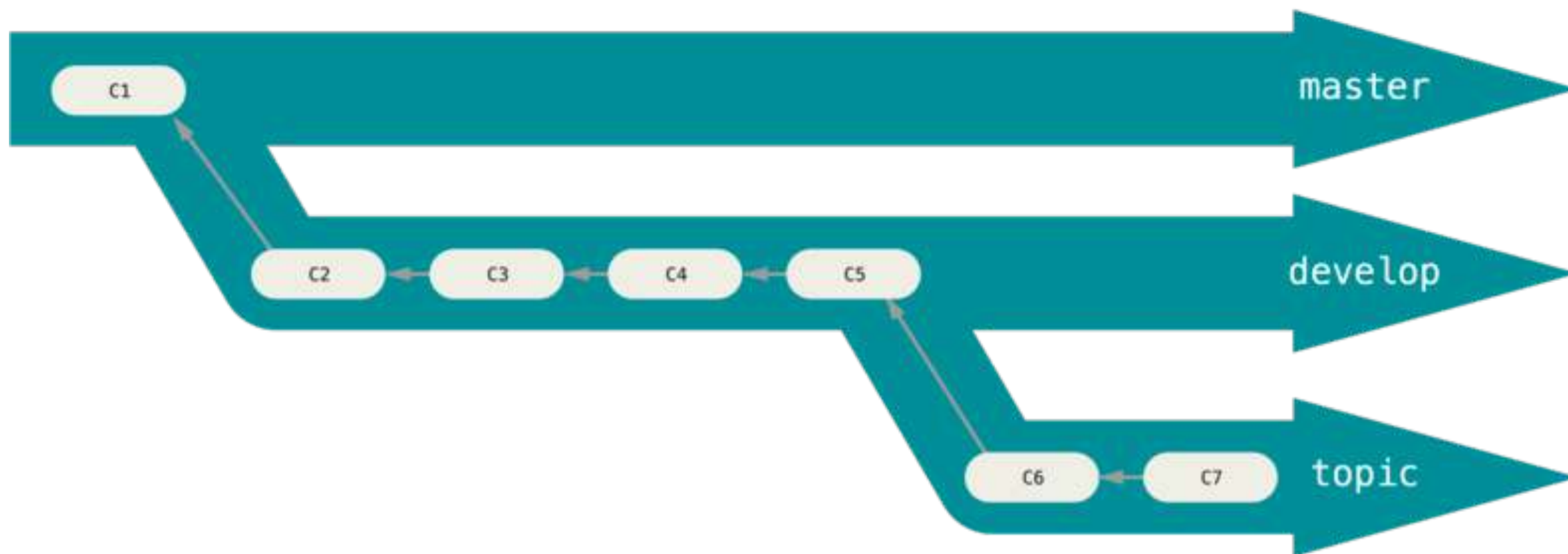
Слияние веток



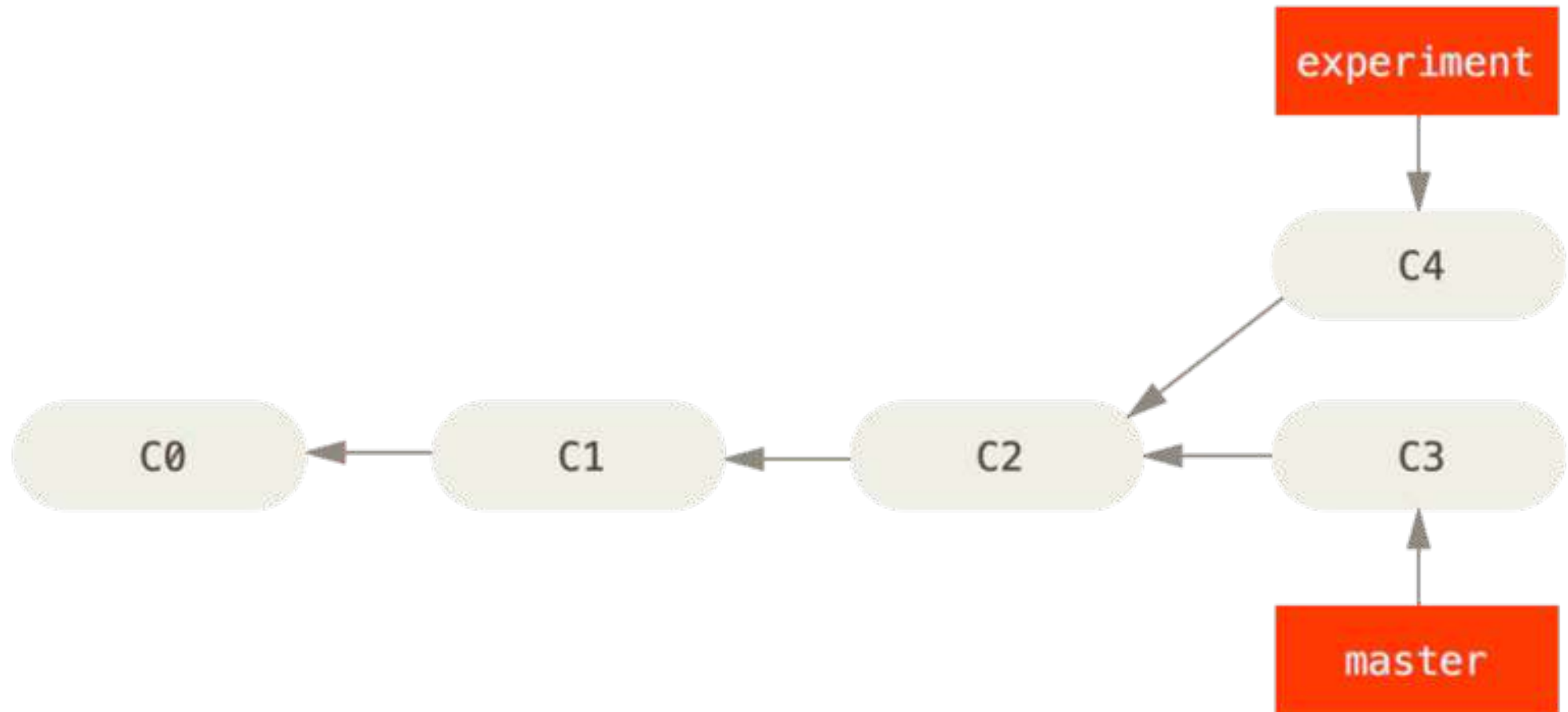
Слияние веток



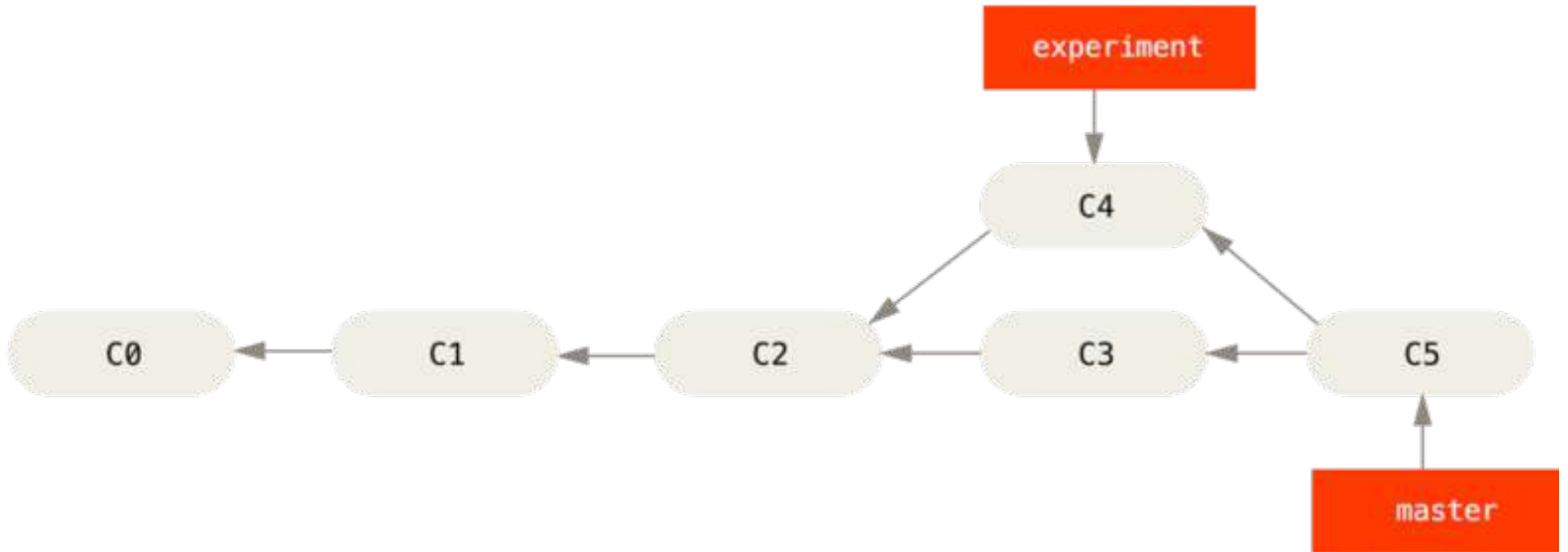
Частый flow



Перебазирование веток



Перебазирование веток (это merge)



Перебазирование веток (это rebase)

