# **Разрешение конфликта вручную и через vimdiff**

В этой теме поговорим о **конфликтах слияния**. Они могут случиться при слиянии веток (merge), если один и тот же файл был изменён в обеих ветках. Git не может сам решить, какая версия файла «правильная», а какая нет.

Такие конфликты нередко возникают в командной работе. В этом уроке покажем, как разрешать их вручную и через консоль.

### **Подготавливаем репозиторий и создаём конфликт**

Подготовьте новый Git-репозиторий с помощью следующего набора команд.



То же самое одной командой — скопируйте и вставьте в консоль

Обратите внимание: все команды объединены через символ &&, а строки — через слеш (\). Слеш в конце консольной команды позволяет продлить её на следующую строку.

В примере мы используем ветку main как основную. Возможно, вам понадобится поменять её на master.

mkdir git-conflict && cd git-conflict && git init && \

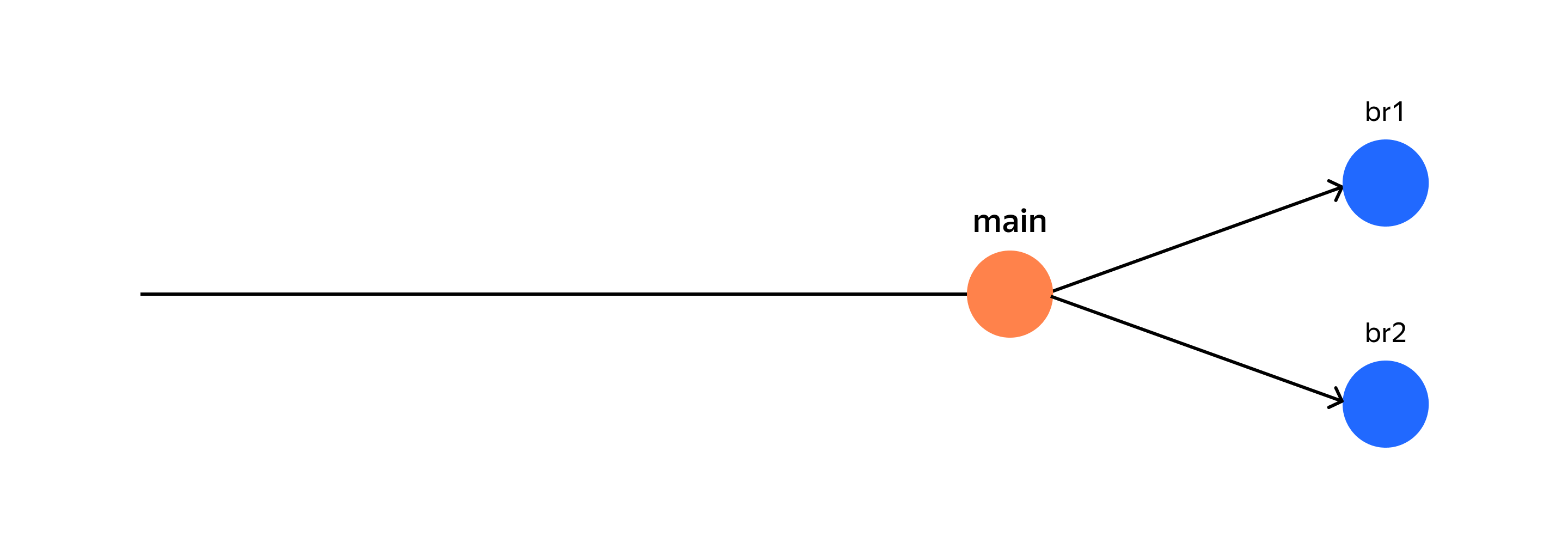
echo 'main version' > readme.md && git add . && git commit -m 'main' && \

git checkout -b br1 && echo 'version 1' > readme.md && git add . && git commit -m 'v1' && \

git checkout main && \

git checkout -b br2 && echo 'version 2' > readme.md && git add . && git commit -m 'v2'

После выполнения этих команд будет создан репозиторий с тремя коммитами: первый добавляет текст main version, а два других — текст version 1 и version 2. Обе ветки отходят от main и изменяют один и тот же файл. При попытке объединить все три ветки получится конфликт.



Добавьте в основную ветку (main или master) ветку br1. Коммиты этих веток можно выстроить в одну линию, поэтому слияние будет выполнено в режиме fast-forward.

$ git checkout main && git merge br1

# ваша основная ветка может называться master

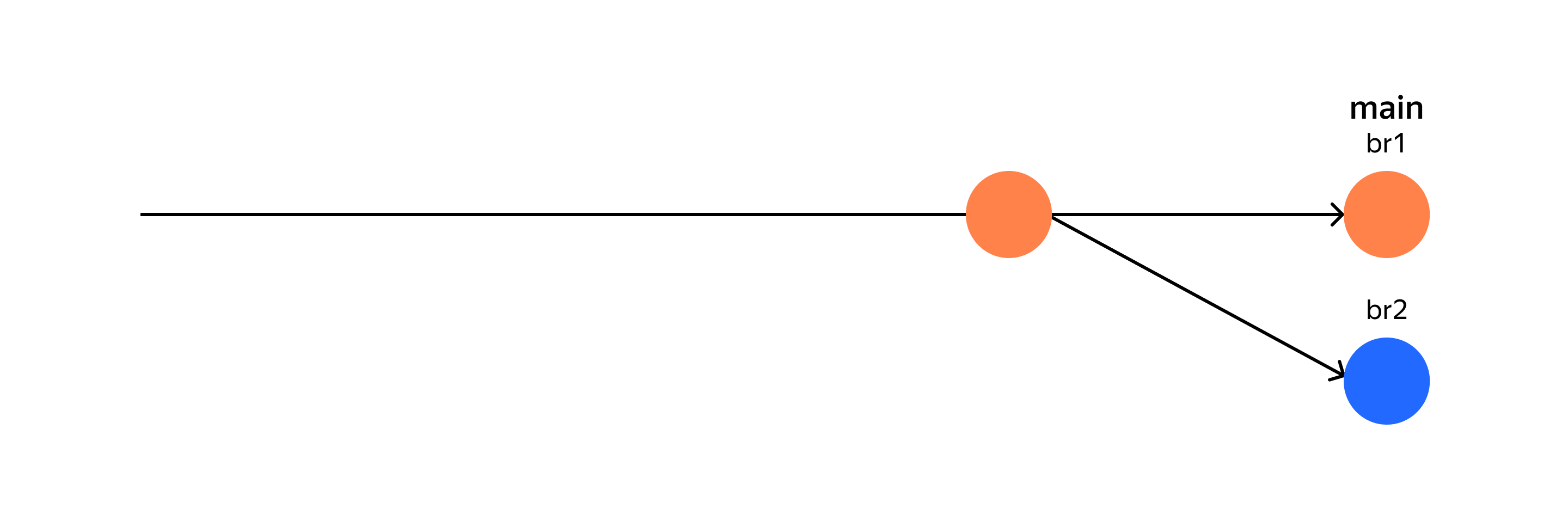
Updating 5f57121..c354ae5

Fast-forward

readme.md | 2 +-

1 file changed, 1 insertion(+), 1 deletion(-)

Теперь в файле readme.md содержится текст version 1, а текущее состояние веток будет такое.



Настало время создать конфликт! Для этого убедитесь, что находитесь в основной ветке, а затем выполните git merge.

$ git checkout main && git merge br2

# ваша основная ветка может называться master

Auto-merging readme.md

CONFLICT (content): Merge conflict in readme.md

Automatic merge failed; fix conflicts and then commit the result.

Поздравляем: конфликт получен!

### **Разрешаем конфликт вручную**

Когда Git выявляет конфликт, он помечает проблемные файлы и останавливает процесс слияния.

Если выполнить git status в процессе «мёржа», файлы с маркерами конфликтов будут отображаться в особой категории **Unmerged paths** (англ. «не соединённые пути») с текстом **both modified** (англ. «оба изменены»). Вывод будет такой.

$ git status

On branch main

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: readme.md

При попытке объединить ветки или применить изменения из удалённого репозитория Git добавит в файлы специальные маркеры конфликта. Убедитесь в этом. Откройте файл readme.md в графическом интерфейсе или выполните cat readme.md. Вы увидите следующее.

$ cat readme.md

<<<<<<< HEAD

version 1

=======

version 2

>>>>>>> br2

Git разметил файл. Получившиеся секции содержат изменения из каждой ветки:

* Текст между <<<<<<< HEAD и ======= указывает на изменения, которые находятся в HEAD — в данном случае это ветка main. Здесь окажутся только те строки, в которых есть конфликт.
* Текст между ======= и >>>>>>> br2 показывает на изменения, которые находятся в ветке br2.

Чтобы разрешить конфликт вручную, нужно открыть файл и выбрать, какие изменения оставить, а какие отбросить. Для этого следует удалить все маркеры и ненужные изменения и оставить нужные. После разрешения конфликта файлы будут отмечены как решённые. Можно продолжить процесс слияния или выполнить коммит изменений.

Допустим, нужно оставить текст version 2. Откройте файл readme.md и удалите все маркеры конфликтов, а также строку version 1. Должно получиться следующее.

version 2

Подготовьте изменения к сохранению и сделайте коммит.

$ git add . && git commit --no-edit

Готово! Вы разрешили конфликт вручную и создали коммит слияния. Теперь в файле readme.md содержится текст version 2. Дерево коммитов будет выглядеть так.

