# \_УП\_9

# Работа с Git

Команда git branch делает несколько больше, чем просто создаёт и удаляет ветки. При запуске без параметров, можно получить простой список имеющихся веток (рисунок 1). Символ \*, стоящий перед веткой master указывает на ветку, на которую указывает HEAD).

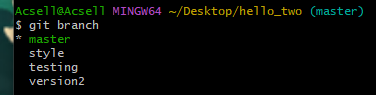


Рисунок 1 - Список существующих веток

Чтобы посмотреть последний коммит на каждой из веток, необходимо выполнить команду git branch -v (рисунок 2).

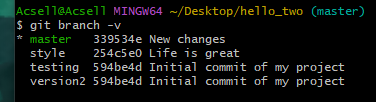


Рисунок 2 - Список веток с последними коммитами

Опции --merged и --no-merged могут отфильтровать этот список для вывода только тех веток, которые слиты или ещё не слиты в текущую ветку (рисунки 3-4).

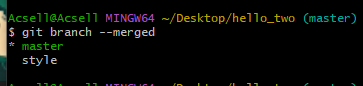


Рисунок 3 - Список веток слитых с текущей

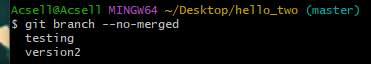


Рисунок 4 - Список веток не слитых с текущей

Затем следует удалить ветку testing (рисунок 5). При наличии ошибок для удаления можно использовать параметр -D.

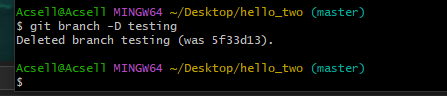


Рисунок 5 - Удаление ветки

Для получения списка удалённых веток и дополнительной информации используется команда git remote show (рисунок 6).

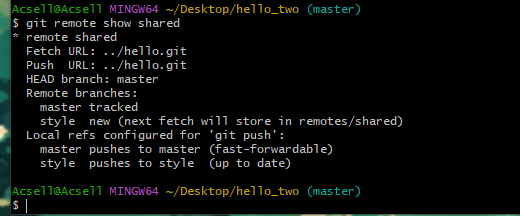


Рисунок 6 - Просмотр удаленных веток

Для отправления изменений на удалённый сервер используется команда git push <remote> <branch> (рисунок 7).

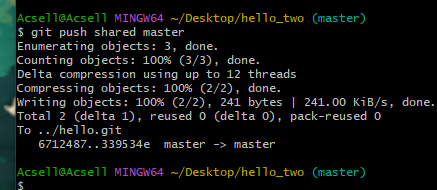


Рисунок 7 - Отправка изменений

Далее при получении обновлений с сервера будет показана ссылка на удаленную ветку (рисунок 8).



Рисунок 8 - Выполнение команды git fetch

При необходимости можно создать локальную ветку на основе удаленной (рисунок 9).

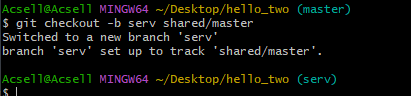


Рисунок 9 - Создание ветки на основе удаленной ветки

Для удаления веток на удаленном сервере используется команда, показанная на рисунке 10.

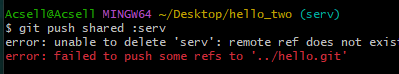


Рисунок 10 - Удаление ветки на сервере

Простой способ выполнить слияние двух веток – это команда merge. Другой способ – использование команды rebase, что означает перебазирование (рисунок 11). Это работает следующим образом: берётся общий родительский снимок двух веток (текущей, и той, поверх которой вы выполняете перебазирование), определяется дельта каждого коммита текущей ветки и сохраняется во временный файл, текущая ветка устанавливается на последний коммит ветки, поверх которой выполняется перебазирование, а затем по очереди применяются дельты из временных файлов.

Далее после этого надо переключиться на ветку master и выполнить перемотку.

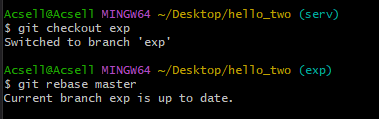


Рисунок 11 - Перемещение изменений

При наличии ответвления от ветки (сначала было ответвление на ветку se, а затем от нее на ветку cl), чтобы переместить изменения можно осуществить действия, показанные на рисунках 12-15.

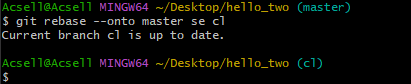


Рисунок 12 - Перемещение изменений с параметром onto

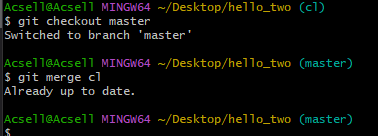


Рисунок 13 - Слияние веток master и cl



Рисунок 14 - Перемещение изменений

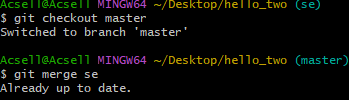


Рисунок 15 - Слияние веток master и se

После этого перемещение будет осуществлено и ветки можно удалить (рисунок 16).

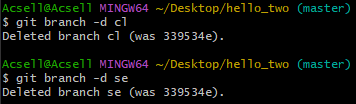


Рисунок 16 - Удаление веток cl и se