

Семинар #2: git, продолжение. Практика.

Для сдачи этого и будущих ДЗ вам нужно создать репозиторий на GitLab (если он ещё не создан) под названием `devtools-homework`. Структура репозитория должна иметь вид:

```
├── seminar2_more_git/  
│   ├── 01.sh  
│   ├── 02.sh  
│   └── ...  
└── ...
```

Подготовка. Генерация репозитория, который будет использоваться в задачах.

Для решения следующих задач вам потребуется создать простой репозиторий, состоящий из четырёх коммитов, с помощью `bash`-скрипта. Для этого сделайте следующее:

1. Зайдите в репозиторий `mipt-hsse.gitlab.yandexcloud.net/v.biryukov/devtools_course`, перейдите в папку `seminar2_git_advanced/practice`, откройте файл `create_animals.sh` и скачайте его.
2. Скопируйте скрипт в папку, в которой вы будете его запускать. Убедитесь, что папка, куда вы копируете скрипт, не является частью другого репозитория, а также не содержит внутри себя папку `animals`.
3. Откройте терминал и зайдите в папку, содержащую скрипт

```
$ cd имяпапки
```

4. Добавьте скрипту права на исполнение

```
$ chmod +x create_animals.sh
```

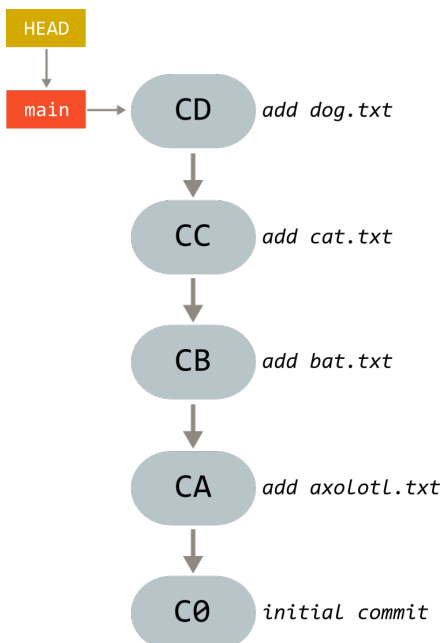
5. Запустите скрипт, он создаст папку `animals` с репозиторием внутри

```
$ ./create_animals.sh
```

6. Перейдите в папку `animals` и убедитесь, что репозиторий создан корректно, напечатав дерево его коммитов:

```
$ cd animals
```

```
$ git log --oneline --graph --all
```



Задача 1. Отмена индексации

Предположим, что вы создали новый файл и добавили его в область индексирования, используя команды:

```
$ touch emu.txt
$ git add emu.txt
```

Однако, затем передумали и хотите убрать этот файл из области индексирования. Выполните команду, которая бы убирала этот файл из области индексирования, но оставляла бы файл в рабочей папке. Добавлять новые коммиты в этой задаче нельзя.

Задача 2. Отмена коммита

Предположим, что вы создали новый файл и добавили его в область индексирования и в локальный репозиторий:

```
$ touch emu.txt
$ git add emu.txt
$ git commit -m "CE: add emu.txt"
```

Однако, затем передумали и хотите отменить коммит. При этом вы хотите, чтобы файл `emu.txt` продолжал находиться и в рабочей папке и в области индексирования. Выполните команду, которая отменит коммит, оставив рабочую папку и область индексирования неизменными.

Задача 3. Отмена reset

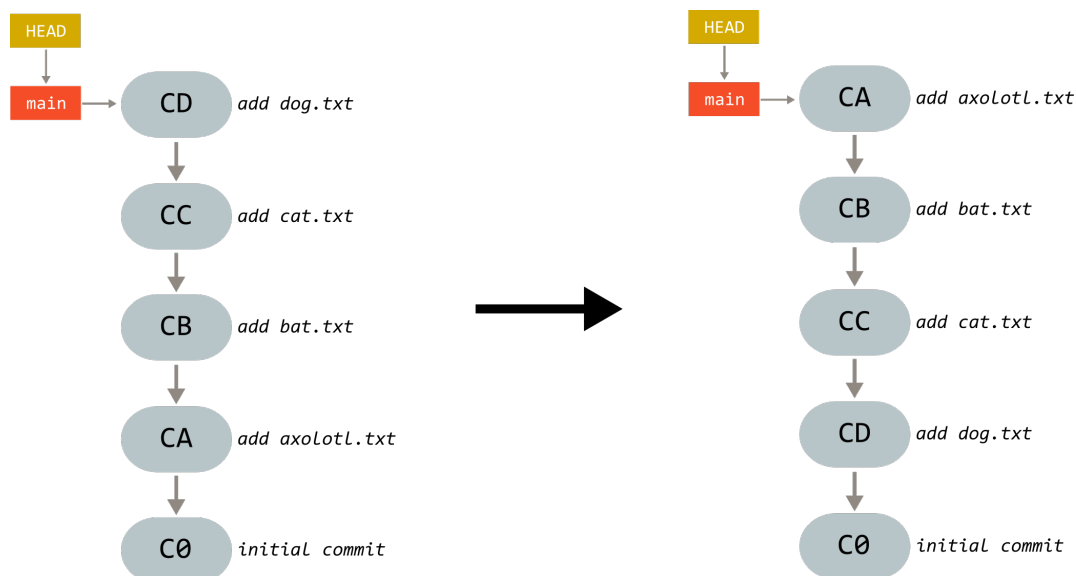
Предположим, что вы решили удалить все коммиты, кроме первого и, находясь на коммите `CD` (*add dog*), выполнили:

```
$ git reset --hard HEAD~4
```

Но потом передумали, и решили вернуть всё обратно. Что для этого нужно сделать?

Задача 4. Обратить коммиты

В этом задании вам нужно обратить коммиты как это показано на изображении. При этом создавать файлы вручную нельзя, можно только использовать команды `git`.

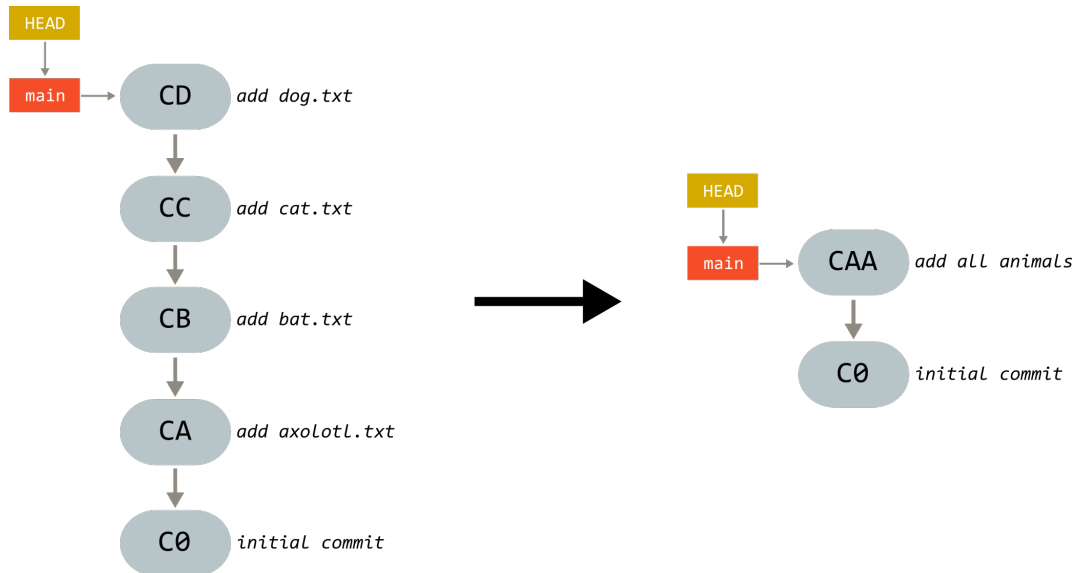


Решите эту задачу двумя способами:

1. Создав новую ветку на начальном коммите и добавляя коммиты с помощью `git cherry-pick`. После этого нужно будет удалить старую ветку и переименовать новую.
2. Используя интерактивное перебазирование.

Задача 5. Объединить коммиты

Пересоздайте репозиторий с помощью скрипта `create_animals.sh`. В этом задании вам нужно объединить все коммиты в один как это показано на изображении. При этом создавать файлы вручную нельзя, можно только использовать команды `git`.

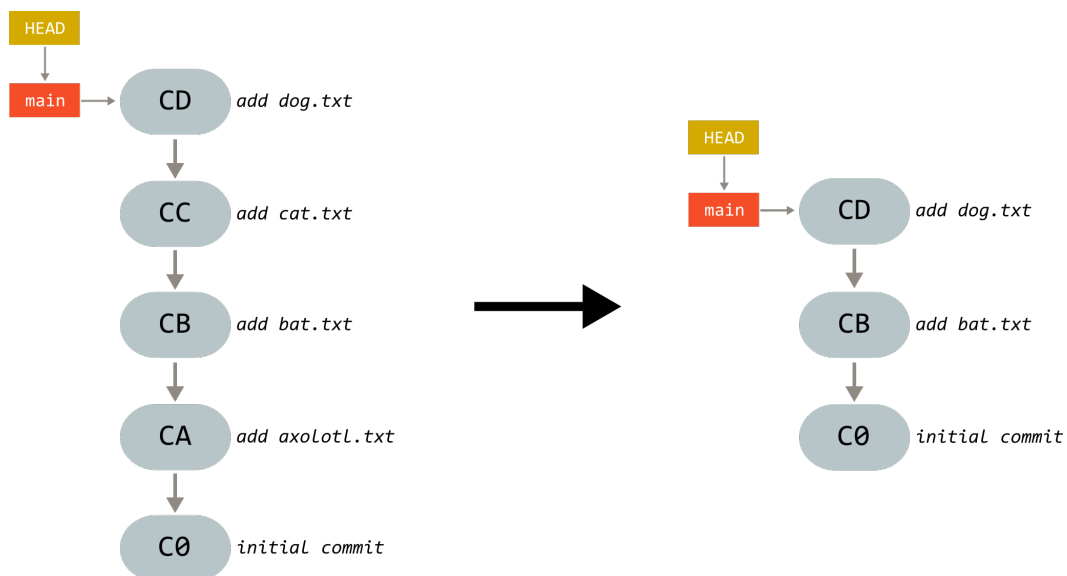


Решите эту задачу двумя способами:

1. Используя `reset`.
2. Используя интерактивное перебазирование.

Задача 6. Удалить коммиты

Пересоздайте репозиторий с помощью скрипта `create_animals.sh`. В этом задании вам нужно удалить два коммита, как это показано на изображении. При этом создавать или удалять файлы вручную нельзя, можно только использовать команды `git`.

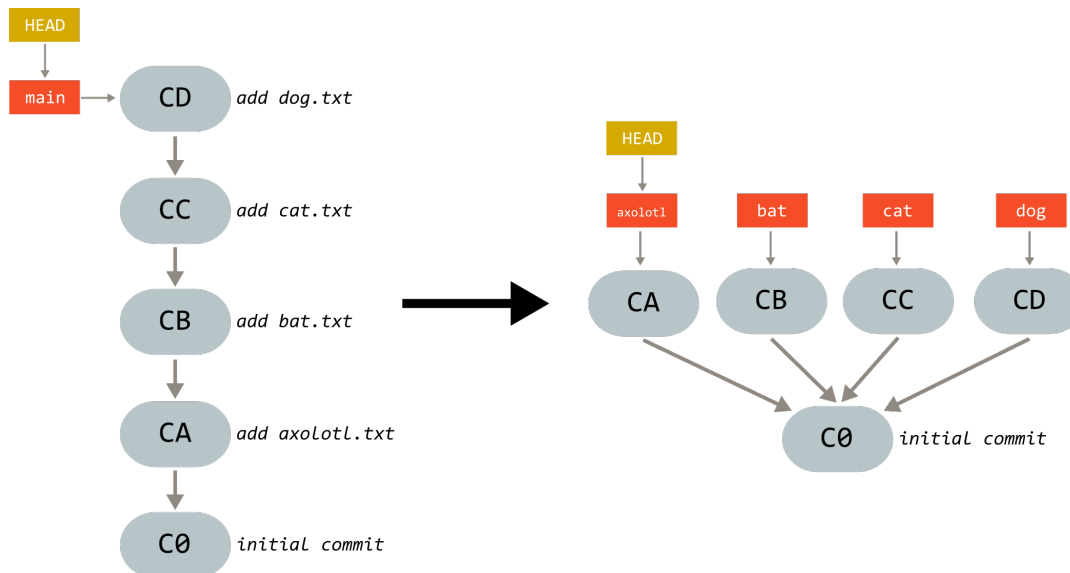


Решите эту задачу двумя способами:

1. Используя `cherry-pick`.
2. Используя интерактивное перебазирование.

Задача 7. Перераспределить коммиты

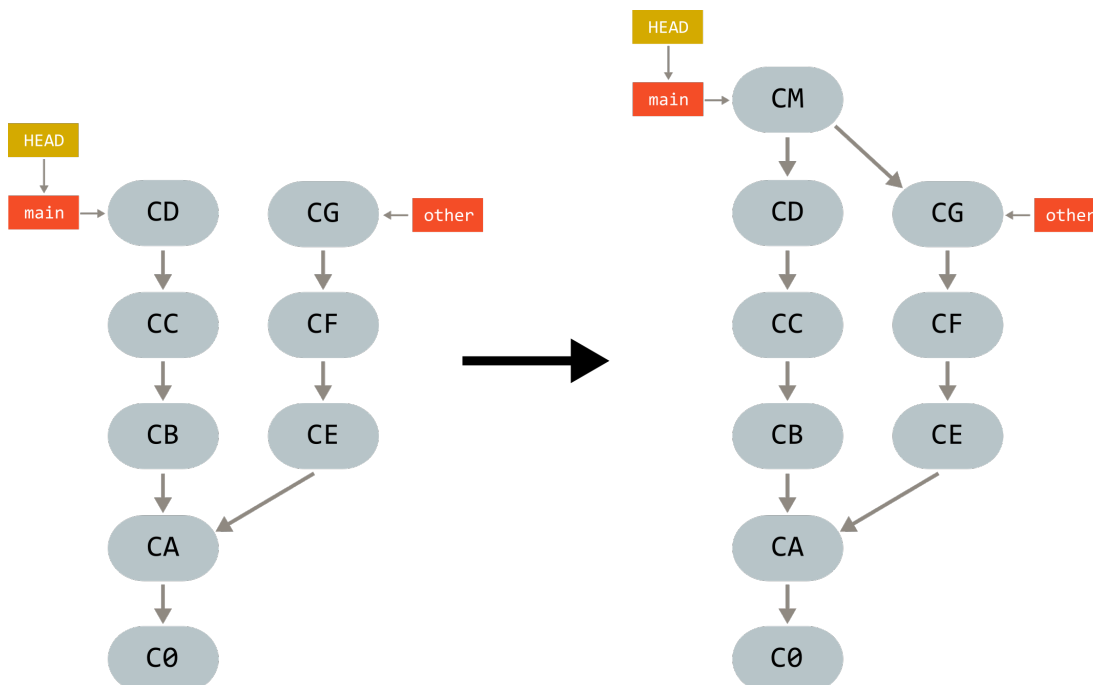
Пересоздайте репозиторий с помощью скрипта `create_animals.sh`. В этом задании вам нужно перераспределить коммиты в новые ветки, как это показано на изображении. При этом создавать или удалять файлы вручную нельзя, можно только использовать команды `git`.



Задача 8. Отмена слияния

Создайте новый репозиторий с помощью скрипта `create_branch_animals.sh` (его можно скачать оттуда же, где находился предыдущий скрипт). В этом репозитории будет 2 ветки: `main` и `other`. Предположим, что вы сделали слияние с помощью команд:

```
$ git switch main
$ git merge other -m "CM: merging"
```

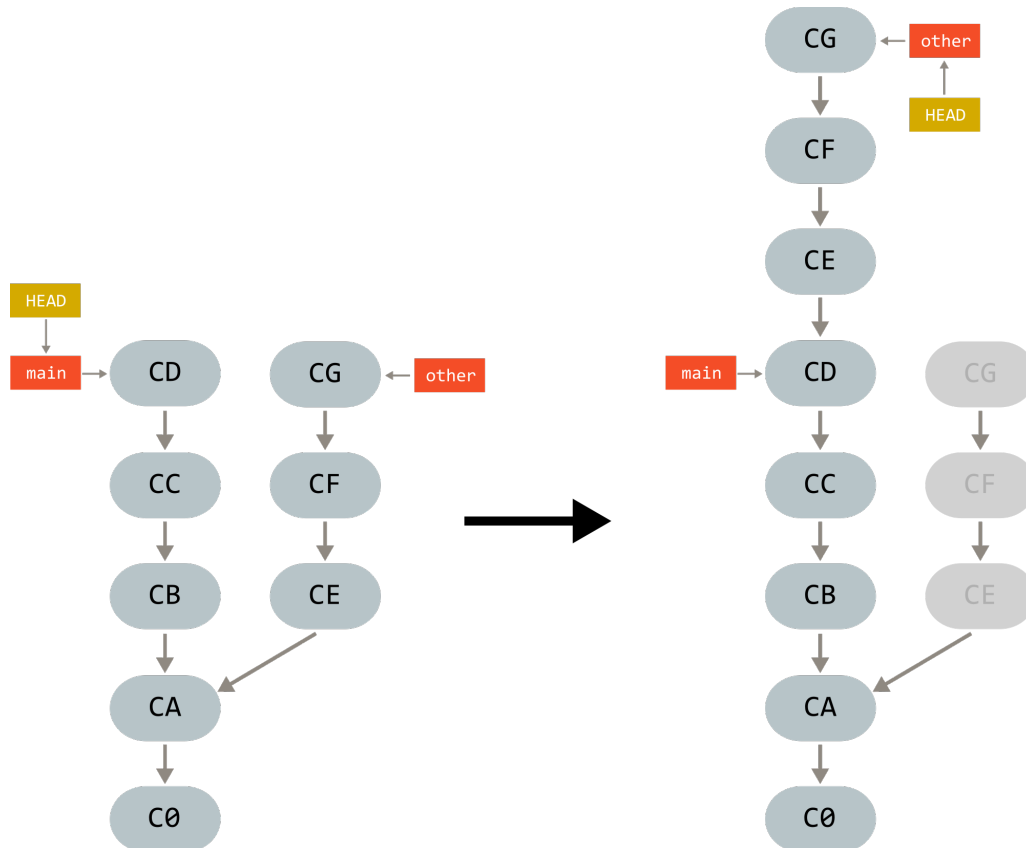


Однако потом передумали и хотите отменить слияние. Напишите команды `git`, которые бы отменяли слияние и возвращали бы всё как прежде.

Задача 9. Отмена перебазирования

Создайте новый репозиторий с помощью скрипта `create_branch_animals.sh`. В этом репозитории есть 2 ветки: `main` и `other`. Предположим, что вы сделали перебазирование ветки `other` на ветку `main` с помощью команд:

```
$ git switch other
$ git rebase main
```



Однако потом передумали и хотите отменить перебазирование. Напишите команды `git`, которые бы отменяли перебазирование и возвращали бы всё как прежде.

Задача 10. Отмена push

Для этой задачи вам нужно:

1. Создать новый локальный репозиторий на вашем компьютере с помощью скрипта `create_animals.sh`.
2. Создать новый пустой репозиторий на GitLab.
3. Скопировать ссылку на GitLab репозиторий (протокол SSH).
4. Добавить новый remote на вашем локальном репозитории.

```
$ git remote add имя ссылка
```

Имя можно выбрать любое, но часто выбирают `origin`.

5. Проверьте, что новый remote добавился:

```
$ git remote -v
```

6. Отправьте ветки `main` и `other` вашего локального репозитория на удалённый репозиторий GitLab.

```
$ git push -u origin main other
```

Предположим, что вы создали новый файл и добавили его в область индексирования, в локальный репозиторий, а затем и в удалённый репозиторий.

```
$ touch emu.txt
$ git add emu.txt
$ git commit -m "add emu.txt"
$ git push
```

Однако затем передумали и хотите отменить коммит. Вы хотите, чтобы изменения пропали везде – как на вашем компьютере, так и в последнем состоянии удалённого репозитория. Решите эту задачу двумя способами:

- Используя `git push`.
- Используя `git revert`.

Задача 11. Форк

Зайдите на GitLab и сделайте форк репозитория `mipt-hsse.gitlab.yandexcloud.net/v.biryukov/branch_animals`. Клонировать этот репозиторий на свой компьютер. Создайте новую ветку на коммите `HEAD`. Добавьте в этой ветке новый коммит, содержащий файл с новым животным (на ваш выбор). Отправьте изменения на ваш форк. Сделайте `merge request`, чтобы добавить изменения в изначальный репозиторий (`v.biryukov/branch_animals`).

Задача 12. Pre-commit hook

Напишите `pre-commit` хук, который будет проверять, что все файлы имеют расширение `.txt`. Проверьте работу этого хука, закомитив файл с другим расширением.

Задача 13. Тэги

Создайте 2 новых тэга в репозитории `branch_animals`: один на коммите `CA` – `v0.1.0`, второй на коммите `CD` – `v0.1.0`. Проверьте, что тэги созданы:

```
git log --oneline --graph --all
```

Отправьте теги на удалённый сервер.