

Системы контроля версий на примере Git

Соболева Дарья

ВМК МГУ
Кафедра ММП
Курс «Практикум на ЭВМ» (317)

14 ноября 2016 г.

О контроле версий

Система контроля версий — это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.

Необходимость в системах контроля версий

- ▶ Хранение нескольких версий одной программы.
- ▶ Командная разработка.
- ▶ Восстановление после ошибок.
- ▶ Ветвление и управление релизами.

Схема работы Git

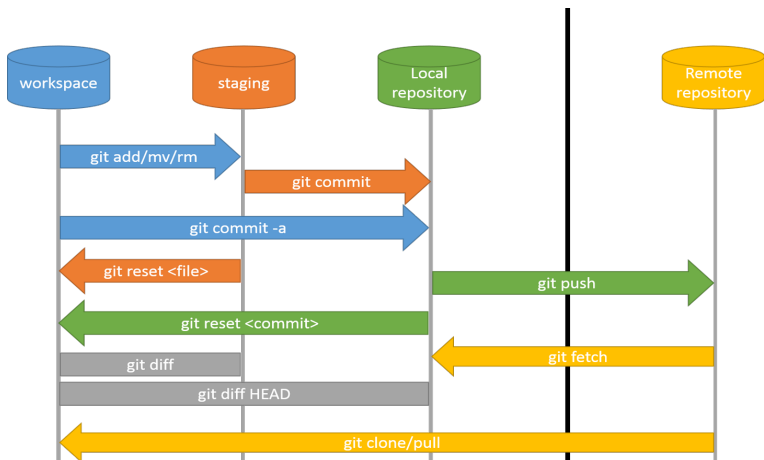


Рис.: Схема работы Git

Создание репозитория

- Создание репозитория в существующем каталоге:

```
$ git init
```

- Клонирование существующего репозитория:

```
$ git clone [url]
```

Запись изменений в репозиторий

- Добавление изменённых файлов в индекс:

```
$ git add file [, file]
```

- Фиксация изменений:

```
$ git commit
```

- Откат к предыдущему состоянию проекта:

```
$ git reset --hard HEAD^
```

Команды слияния веток (1)

```
$ git merge
```

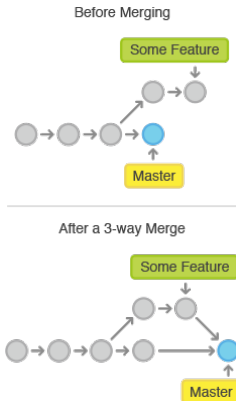


Рис.: Команда merge

Команды слияния веток (2)

```
$ git rebase
```

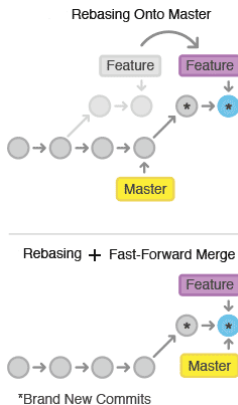


Рис.: Команда rebase

Работа с удалёнными репозиториями

- Получение данных из удалённых проектов:

```
$ git fetch [remote_server_name]
```

- Обновление удалённого репозитория (коммит):

```
$ git push [remote_server_name]
```

Почему Git

- ▶ Все локально.
- ▶ Контроль.
- ▶ Эффективность.
- ▶ Безопасность.
- ▶ Популярность и жизнеспособность.
- ▶ Наличие качественных образовательных материалов.

Используйте Git

«Почему вы ещё не пользуетесь Git?

Да, да, это я к вам: что же вас сдерживает от использования контроля версий на базе Git?» - Хабрахабр

```
$ Thanks for your attention!
```

Литература

- 🌐 Почему Git <https://habrahabr.ru/post/104198/>
- 🌐 6 мифов, мешающих разработчикам использовать Git <https://habrahabr.ru/post/231839/>
- 🌐 Достаточно Git-а, чтобы быть (менее) опасным <https://habrahabr.ru/post/268951/>
- 🌐 Pro Git Book <https://git-scm.com/book/ru/v1>