

# Системы контроля версий на примере Git

Каюмов Эмиль

ММП ВМК МГУ

Курс «Практикум на ЭВМ» (317)

27 октября 2015

# Необходимость в системах контроля версий

- Хранение нескольких версий одной программы (возможность отката к предыдущей версии).
- Совместная работа группы разработчиков (объединение результатов).
- Хранение истории разработки с комментариями (поиск виновного в ошибках).
- Реализация альтернативных вариантов программы (ветвление).

# Схема работы с Git (1)

- Создаём новый репозиторий в текущей папке:

```
# git init
```

- Или копируем на свой компьютер имеющийся:

```
# git clone http://url/project.git
```

- Теперь можем создавать или редактировать файлы в локальном репозитории.

## Схема работы с Git (2)

- Зафиксируем сделанные изменения.
- Добавляем изменённые файлы в индекс (временное хранилище):

```
# git add file1 file2
```

- Делаем коммит совершенных изменений:

```
# git commit -m "commit comment"
```

- Теперь изменения зафиксированы в локальном репозитории.
- Вернём состояние проекта на 3 коммита назад:

```
# git reset --hard HEAD-3
```

# Схема работы с Git (3)

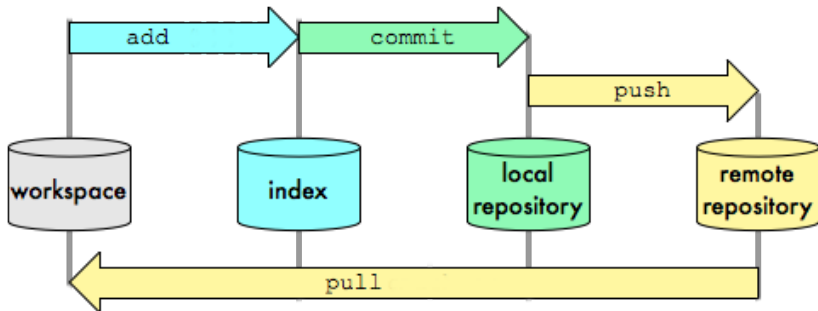
- Обновим локальный репозиторий (загрузим последние изменения):

```
# git pull
```

- Обновим удалённый репозиторий (отправим коммиты):

```
# git push
```

# Упрощённая схема



- Создан Линусом Торвальдсом в 2005 для управления разработкой ядра Linux.
- Git – распределённая система контроля версий (не требуется постоянное соединение с сервером).
- Работа с изменениями, а не с файлами. Единицей обработки является набор изменений (патч).

# Преимущества Git

- Гибкая система ветвления проектов и слияния веток между собой.
- Наличие локального репозитория позволяет вести полноценный локальный контроль версий и заливать в главный репозиторий только законченные исправления.
- Высокая скорость работы и эффективное использование дискового пространства по сравнению с другими системами контроля версий.
- Широкая распространенность, доступность.
- Гибкость системы (создание специализированных систем контроля или пользовательских интерфейсов на базе Git).
- Универсальный сетевой доступ с использованием протоколов http, ftp, rsync, ssh и других.



- Отслеживание изменения не отдельных файлов, а только всего проекта целиком (неудобно при множестве несвязных файлов).
- Большие временные затраты при начальном (первом) создании репозитория и синхронизации его с другими разработчиками (требуется скопировать на локальный компьютер весь репозиторий).

- 🌐 Команды Git <http://www.calculate-linux.org/main/ru/git>
- 🌐 Git Book <http://git-scm.com/book/en/v2>
- 🌐 Почему Git <http://habrahabr.ru/post/104198/>